Consistent Consequence for Boolean Equation Systems

M.W. Gazda and T.A.C. Willemse

Eindhoven University of Technology
Eindhoven, The Netherlands

Abstract. Inspired by the concept of a consistent correlation for Boolean equation systems, we introduce and study a novel relation, called *consistent consequence*. We show that it can be used as an approximation of the solution to an equation system. For the closed, simple and recursive fragment of equation systems we prove that it coincides with *direct simulation* for parity games. In addition, we show that deciding both consistent consequence and consistent correlations are coNP-complete problems, and we provide a sound and complete proof system for consistent consequence. As an application, we define a novel abstraction mechanism for parameterised Boolean equation systems and we establish its correctness using our theory.

1 Introduction

Boolean equation systems [9] have been studied extensively in the context of software and hardware verification, in which the equation systems appear naturally as the end result of encodings of model checking problems, including the modal μ-calculus problem [9] and data and real-time model checking problems [15]; but also as a result of process equivalence checking problems [11]. The encoded verification problems can be answered by solving the associated equation system, which is known to be in NP $\cap$ coNP.

The Boolean equation systems originating from verification problems tend to be rather large. However, so called *parameterised Boolean equation systems* [10, 6] can be used to concisely describe the Boolean equation systems, using a combination of data, parameterised recursion and first-order quantification. Subsequently generating the Boolean equation systems from such parameterised Boolean equation systems, however, leads to problems akin to the infamous state-space explosion. In an attempt to side-step the latter phenomenon, transformations that simplify parameterised Boolean equation systems have been studied, see *e.g.* [13]; such transformations are reasonably cheap, but can be extremely effective at reducing the size of the underlying Boolean equation system.

Establishing the soundness of a given transformation has long been a rather tiresome affair; only recently, a proof methodology has been devised that avoids the need for lengthy proofs. The method relies on the identification of a suitable *consistent correlation* [14] between two (parameterised) Boolean equation systems. While the decidability of such consistent correlations was addressed for a fragment of Boolean equation systems, the decidability and the computational complexity for the general setting of Boolean equation systems remained open.

An inconvenience of consistent correlations is that they only serve to reason about transformation that are solution preserving and reflecting. The technique falls short in situations in which one wishes to devise more aggressive transformations that allow us to under- or over-approximate the solution to a given (parameterised) Boolean equation system. Compared to solution preserving transformations, such transformations offer the prospect of even more powerful reductions. To accommodate the development of such transformations, we define and study a *preorder* that is tightly related to the concept of consistent correlation. The resulting preorder is called *consistent consequence*. In addition, we study its computational complexity and its relation to concepts known in the setting of Boolean equation systems and in related settings. Our theoretical contributions can be summarised as follows:

- We establish the relation between consistent consequence, consistent correlation and the concept of a solution to an equation system;
- We link consistent consequence for the fragment of closed Boolean equation systems in simple and recursive form to *direct simulation* for *parity games* [12], providing a graph-based, operational view on the concept;
- We prove that deciding consistent consequence and consistent correlation are both coNP-complete problems, answering the open problem raised in [14];
- We give a sound and complete proof system for consistent consequence and claim the soundness and completeness of the proof system for consistent correlation, conjectured in [14].

Given the notationally more involved setting of parameterised Boolean equation systems, we have opted to study the concept of consistent consequence in the more accessible framework of Boolean equation systems. Lifting the concepts to the richer setting, following [14], is interesting but routine.

We conclude our paper by defining a novel abstraction mechanism for (parameterised) Boolean equation systems, the soundness of which is easily established by an appeal to our new theory. The abstraction mechanism allows one to solve parameterised Boolean equation systems that could not be solved before.

Outline We give a cursory overview of the Boolean equation system framework in Section 2. In Section 3, we define and study our notion of consistent consequence, and we briefly analyse its computational complexity. The correspondence to direct simulation for parity games is addressed in Section 4. In Section 5, we present our proof system for consistent consequence and state the soundness and completeness of the proof system for consistent correlation, conjectured in [14]. We apply our theory by defining a novel manipulation on parameterised Boolean equation systems, and prove its correctness in Section 6. We wrap up with some ideas for future work in Section 7.

2 Preliminaries

Boolean equation systems are finite sequences of fixed point equations, in which the right-hand sides of the equations are proposition formulae. We assume that the reader has some familiarity with fixed point theory and Boolean equation systems; for an excellent, in-depth account on the latter, we refer to [9].

Definition 1. *A* Boolean equation system (BES) $\mathcal{E}$ *is defined by the following grammar:*

$$\begin{aligned} \mathcal{E} \quad &::= \epsilon \mid (\nu X = f)\ \mathcal{E} \mid (\mu X = f)\ \mathcal{E} \\ f, g &::= \top \mid \bot \mid X \mid f \wedge g \mid f \vee g \end{aligned}$$

The empty BES is denoted ϵ; X is a proposition variable taken from a countable set $\mathcal{X}$ of proposition variables; f, g are proposition formulae.

From hereon, whenever we write σ, we mean an arbitrary fixed point sign, *i.e.*, either μ or ν. As a notational convention, we write f_X when we refer to the right-hand side proposition formula in the equation for X: we have $\sigma X = f_X$.

Let $\mathcal{E}$ be an arbitrary equation system. We denote the set of *bound* proposition variables of $\mathcal{E}$ by $\mathsf{bnd}(\mathcal{E})$; that is, $\mathsf{bnd}(\mathcal{E})$ contains those variables at the left-hand side of the equations in $\mathcal{E}$. We only consider equation systems in which all equations have unique left-hand side variables. Proposition variables occurring in a proposition formula f are collected in the set $\mathsf{occ}(f)$; by extension, $\mathsf{occ}(\mathcal{E})$ contains variables occurring in the right-hand side of any of the equations in $\mathcal{E}$.

Whenever all occurring variables are bound in an equation system, it is said to be a *closed* equation system. An equation system is in *simple form* [1] if none of the right-hand sides of the equations that occur in the equation system contain both $\wedge$- and $\vee$-operators. If none of an equation system's right-hand side formulae contain the constants $\top$ and $\bot$, the system is in *recursive form.*

To each *bound* variable X of $\mathcal{E}$, we associate a *rank* $\mathsf{rank}_{\mathcal{E}}(X)$, which is defined as $\mathsf{rank}_{\mathcal{E}}(X) = \mathsf{block}_{\nu,X}(\mathcal{E})$:

$$\mathsf{block}_{\sigma,X}((\sigma' Y = f_Y)\mathcal{E}) = \begin{cases} 0 & \text{if } \sigma = \sigma' \text{ and } X = Y \\ \mathsf{block}_{\sigma,X}(\mathcal{E}) & \text{if } \sigma = \sigma' \text{ and } X \neq Y \\ 1 + \mathsf{block}_{\sigma',X}((\sigma' Y = f_Y)\mathcal{E}) & \text{if } \sigma \neq \sigma' \end{cases}$$

Informally, the rank of a variable X is the i-th block of like-signed equations, containing X's defining equation, counting from left-to-right and starting at 0 if the first block consists of greatest fixed point signs, and 1 otherwise.

Semantics. Let $\eta{:}\mathcal{X} \to \mathbb{B}$ be a *proposition environment*, assigning Boolean values to proposition variables. The semantics of a proposition formula f is given in the context of an environment η (notation $[\![f]\!]\eta$) and is defined as the standard extension of η to formulae. We write $\eta[X := b]$ for the environment η in which the proposition variable X has Boolean value b and all other proposition variables X' have value $\eta(X')$. The ordering $\sqsubseteq$ on environments is defined as $\eta \sqsubseteq \eta'$ if and only if $\eta(X)$ implies $\eta'(X)$ for all X. For reading ease, we do not formally distinguish between a semantic Boolean value and its representation by $\top$ and $\bot$; likewise, for the operands $\wedge$ and $\vee$.

Definition 2. *The* solution of a BES*, given an environment $\eta{:}\mathcal{X} \to \mathbb{B}$, is inductively defined as follows:*

$$\begin{aligned} [\![\epsilon]\!]\eta &= \eta \\ [\![(\sigma X = f)\ \mathcal{E}]\!]\eta &= \begin{cases} [\![\mathcal{E}]\!](\eta[X := [\![f]\!]([\![\mathcal{E}]\!]\eta[X := \bot])]) & \textit{if } \sigma = \mu \\ [\![\mathcal{E}]\!](\eta[X := [\![f]\!]([\![\mathcal{E}]\!]\eta[X := \top])]) & \textit{if } \sigma = \nu \end{cases} \end{aligned}$$

It is not hard to verify that the order of equations impacts the solution: the equation system $(\mu X = Y)(\nu Y = X)$ has $\bot$ as the solution to both X and Y, whereas the equation system $(\nu Y = X)(\mu X = Y)$ will yield the solution $\top$ for both X and Y.

3 Consistent Consequence

Let R be an arbitrary relation on proposition variables $\mathcal{X}$. We write $X \ R \ X'$ iff $(X, X') \in R$. Let θ be an arbitrary environment; θ is *consistent* with R if for all $X \ R \ X'$, $\theta(X) \Rightarrow \theta(X')$. We denote the set of all environments consistent with R by Θ_R.

Definition 3. *Let $\mathcal{E}$ be an equation system. Let R be a relation on proposition variables; R is a* consistent consequence *on $\mathcal{E}$ if for all equations $\sigma X = f_X$ and $\sigma' X' = f_{X'}$ in $\mathcal{E}$ such that $X \ R \ X'$, we have:*

1. $\mathsf{rank}_{\mathcal{E}}(X) = \mathsf{rank}_{\mathcal{E}}(X')$
2. *for all $\theta \in \Theta_R$, we have $[\![f_X]\!]\theta \Rightarrow [\![f_{X'}]\!]\theta$.*

A proposition variable $X' \in \mathsf{bnd}(\mathcal{E})$ is said to be a consistent consequence *of proposition $X \in \mathsf{bnd}(\mathcal{E})$, denoted $X \lessdot X'$ iff there is some relation R that is a consistent consequence on $\mathcal{E}$, such that $X \ R \ X'$.*

Property 1. For any pair of consistent consequences R, S, their union $R \cup S$ is also a consistent consequence.

As a consequence of the above property, we find that there must be a *largest* consistent consequence relation.

Proposition 1. *The relation $\lessdot$ is the largest consistent consequence. Moreover, $\lessdot$ is a preorder.* □

The theorem below is the main result of this section; it establishes a link between the notion of a consistent consequence and the concept of a solution to an equation system.

Theorem 1. *Let $\mathcal{E}$ be an equation system. Let R be a consistent consequence on $\mathcal{E}$. Then for all $\theta \in \Theta_R$ we have $[\![\mathcal{E}]\!]\theta \in \Theta_R$.*

Proof. The proof proceeds using a combination of induction on the length of the equation system $\mathcal{E}$ and approximation of simultaneous fixed points, relying on Bekič principle to convert the nested fixed points to such simultaneous fixed points. □

The notion of consistent consequence is closely related to the notion of a *consistent correlation*, see [14]. For the sake of completeness, we here recall its definition.

Definition 4. *Let $\mathcal{E}$ be an equation system. Let R be a symmetric relation on proposition variables; R is a* consistent correlation *on $\mathcal{E}$ if for all equations $\sigma X = f_X$ and $\sigma' X' = f_{X'}$ in $\mathcal{E}$ such that $X \ R \ X'$, we have:*

1. $\mathsf{rank}_{\mathcal{E}}(X) = \mathsf{rank}_{\mathcal{E}}(X')$
2. *for all $\theta \in \Theta_R$, we have $[\![f_X]\!]\theta = [\![f_{X'}]\!]\theta$.*

Variables $X, X' \in \mathsf{bnd}(\mathcal{E})$ consistently correlate*, denoted $X \doteqdot X'$, iff there is some consistent correlation R such that $X\ R\ X'$.*

The results below relate consistent consequence and consistent correlation: consistent consequence relates to consistent correlation in the same way as simulation preorder relates to bisimulation in the setting of labelled transition systems.

Proposition 2. *Let $\mathcal{E}$ be an arbitrary equation system.*

1. *The largest consistent correlation on $\mathcal{E}$, denoted $\doteqdot$, is the largest symmetric consistent consequence on $\mathcal{E}$;*
2. *The largest consistent correlation on $\mathcal{E}$, $\doteqdot$ is contained in $\lessdot$ and in $\lessdot^{-1}$.* □

We finish this section with an example that illustrates that consistent correlation $\doteqdot$ is, as can be expected, actually finer than $\lessdot \cap \lessdot^{-1}$.

Example 1. Consider the equation system $\mathcal{E}$ given below:

$$(\mu X_0 = X_4)\ (\mu X_1 = X_0 \vee X_2)\ (\mu X_2 = X_4 \vee X_5)\ (\mu X_3 = X_2)\ (\mu X_4 = X_4)\ (\nu X_5 = X_5)$$

Observe that we have, among others, $X_0 \lessdot X_2$, $X_1 \lessdot X_3$, $X_3 \lessdot X_1$, $X_4 \lessdot X_0$; in particular, this means that X_1 is a consistent consequence of X_3 and *vice versa*. Note also that $X_1 \not\doteqdot X_3$: take, for instance $\theta(X_0) = \top$ and $\theta(X_2) = \bot$, which would be allowed if not $X_0 \doteqdot X_2$. But this follows from the fact that $X_4 \not\doteqdot X_5$, as a result of their ranks. Finally, observe that neither $\lessdot$, nor $\doteqdot$ coincide with the partitioning induced by the solution to the equation system: the solution to both X_0 and X_4 is $\bot$, whereas the solution to X_1, X_2, X_3 and, in particular, X_5 is $\top$; note that X_5 cannot be related to, *e.g.*, X_1, due to a difference in ranks. □

Complexity. In [14], it was shown that deciding $\doteqdot$ for the class of equation systems in simple form requires $O(n \log n)$ time, where n is the number of bound variables in an equation system. The complexity for deciding $\doteqdot$ for arbitrary equation systems so far remained an open problem; both problems are coNP decision problems.

Proposition 3. *Let $\mathcal{E}$ be an arbitrary equation system. The decision problems $X \lessdot Y$ and $X \doteqdot Y$, for $X, Y \in \mathsf{bnd}(\mathcal{E})$ are both in coNP.* □

In fact, both decision problems are coNP-complete problems.

Theorem 2. *Deciding $\lessdot$ and $\doteqdot$ is coNP-complete.*

Proof. The problem can be reduced to the logical consequence problem of [3]. □

The coNP-completeness proof relies on an equation system consisting of two blocks. For equation system consisting of only one block, deciding $\doteqdot$ remains coNP-complete, but $\lessdot$ can be decided in linear time.

Remark 1. For equation systems with either all right-hand sides in *Conjunctive Normal Form* or all in *Disjunctive Normal Form*, both decision problems remain polynomial.

4 Consistent Consequence Generalises Direct Simulation on Parity Games

Equation systems generalise *Parity Games*: two-player, graph-based games [12] with ω-winning conditions. We show that for the fragment of equation systems in simple form, consistent consequence on equation systems coincides with *direct simulation*[1] on parity games. Note that despite the fact that parity games and equation systems are mutually reducible to one another, the correspondence that we establish in this section is non-trivial, given the contrasts between the denotational framework of equation systems and the highly operational characteristics of parity games.

Definition 5. *A* parity game *is a game graph* $\mathcal{G} = \langle V, \to, \Omega, (V_{Even}, V_{Odd})\rangle$*, where* V *is a finite set of vertices partitioned in sets* V_{Even} *and* V_{Odd}*,* $\to \subseteq V \times V$ *is a total set of edges and* $\Omega{:}V \to \mathbb{N}$ *is the priority function. Sets* V_{Even} *and* V_{Odd} *consist of vertices owned by player* even *and* odd*, respectively.*

We forego a formal exposition of the concept of winning vertices in a parity game; for details, we refer to [12]. For our purpose in this section, it suffices to be aware that the set of vertices of a parity game can be partitioned in unique sets W_{Even} and W_{Odd}, representing those vertices in V won by player *even* and those won by player *odd*.[2]

The direct simulation preorder on vertices of a parity game is defined through a game played on an auxiliary graph, called the *simulation game-graph* (*i.e.*, *not* a parity game graph). Given a parity game $\mathcal{G} = (V, \to, \Omega, (V_{\text{Even}}, V_{\text{Odd}}))$, the simulation game is a turn-based game played by players *Duplicator* (D) and *Spoiler* (S) on a game-graph $(V \times V, \to \times \to)$ according to moves adhering to the rules in Table 1. An infinite play $(v_0, w_0)(v_1, w_1) \cdots \in (V \times V)^\omega$ is won by Duplicator if all priorities match along the play, *i.e.*, $\Omega(v_n) = \Omega(w_n)$ for all n; otherwise it is won by Spoiler.

Table 1. Admissible moves in a simulation game-graph.

$(v, w) \in$	1st player	plays on	2nd player	plays on
$V_{\text{Even}} \times V_{\text{Even}}$	S	v	D	w
$V_{\text{Even}} \times V_{\text{Odd}}$	S	v	S	w
$V_{\text{Odd}} \times V_{\text{Even}}$	D	w	D	v
$V_{\text{Odd}} \times V_{\text{Odd}}$	S	w	D	v

Definition 6. *Let* $\mathcal{G} = (V, \to, \Omega, (V_{Even}, V_{Odd}))$ *be a parity game. We say that a vertex* v *is* directly simulated *by vertex* w*, denoted* $v \sqsubseteq_{dir} w$ *if player* Duplicator *(D) has a winning strategy for the simulation game starting in* (v, w)*.*

We have the following relation between vertices won by players *even* and *odd*, and the direct simulation relation, as suggested in [5], and as a consequence of our Theorem 3.

[1] It appears that this notion was never formally defined for parity games, but for a variant in the setting of Büchi automata, see [4], and based on suggestions in [5], it can be reconstructed.

[2] The sets W_{Even} and W_{Odd} should not be confused with V_{Even} and V_{Odd}; these are not related.

Proposition 4. *Let $\mathcal{G} = (V, \to, \Omega, (V_{Even}, V_{Odd}))$ be a parity game. Let $v, w \in V$. If $v \sqsubseteq_{dir} w$ and vertex v is won by player* even*, then so is vertex w.* □

Parity games and equation systems in simple, recursive form are known to coincide: see, *e.g.*, the linear reductions in both directions in [7]. These reductions are such that a vertex in the game is won by player *even* iff the corresponding equation in the equation system has *true* as its solution. Based on these reductions, we have the following result.

Theorem 3. *The preorder $\sqsubseteq_{dir}$ on parity games coincides with the preorder $\lessdot$ on closed equation systems in simple and recursive form.* □

Direct simulation can be computed in polynomial time using the framework for games with Büchi winning conditions, see [4]. Clearly, the same technique applies to computing consistent consequence on closed equation systems in simple recursive form; the extension of this technique to *open* equation systems in simple form is standard.

Example 2. Reconsider the equation system $\mathcal{E}$ from Example 1. Its associated parity game is given below; vertices owned by player *even* are diamond-shaped, whereas vertices owned by player *odd* box-shaped. The priorities are written inside the vertices. One can check that vertex v_{X_1} simulates v_{X_3}, corroborating our earlier claim that $X_1 \lessdot X_3$, and *vice versa*. In a similar vein, we have both $v_{X_0} \sqsubseteq_{dir} v_{X_2}$ and $v_{X_4} \sqsubseteq_{dir} v_{X_0}$. □

$$\begin{array}{l}
(\mu X_0 = X_4) \\
(\mu X_1 = X_0 \vee X_2) \\
(\mu X_2 = X_4 \vee X_5) \\
(\mu X_3 = X_2) \\
(\mu X_4 = X_4) \\
(\nu X_5 = X_5)
\end{array}$$

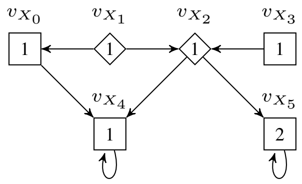

Fig. 1. The equation system $\mathcal{E}$ from Example 1, and its associated parity game.

5 The proof system $\vdash_c$

The definition of a consistent consequence is phrased entirely in terms of the semantics of the artefacts of an equation system. In [14], it was conjectured that the notion of a consistent correlation $\doteqdot$ can be characterised by a proof system that adds only a single coinductive rule to the standard axiomatisation of logical equivalence for negation-free propositional logic. Such an elegant, syntax-based proof system offers an accessible alternative to the semantic definition; this ultimately provides a better understanding of the concept. We vindicate the conjecture in [14], and provide a similar-spirited proof system for consistent consequence.

We write $f \subset g$ to denote that g is a logical consequence of f. Let $\vdash_P$ denote the proof system for logical consequence for negation-free propositional logic, given by the rules in Table 2, save the rules CC and CNT.

Table 2. Proof system for negation-free logical consequence and consistent consequence; α, β, γ represent arbitrary proposition formulae; X, Y are proposition variables; and $\mathit{\Gamma}$ is a context. Furthermore, ς is a substitution mapping proposition variables to proposition formulae; $f\varsigma$ denotes the natural extension of mapping ς from variables to terms. The rules axiomatise associativity (AS), distributivity (DS), absorption (AB), idempotence (ID), supremum (SUP) and infimum (INF) and top (TOP) and bottom (BOT).

Axioms for negation-free propositional logic			
rules of the form $\dfrac{}{\mathit{\Gamma} \vdash \ A}$, where A ranges over the following laws:			
AS1	$\alpha \wedge (\beta \wedge \gamma) \subset (\alpha \wedge \beta) \wedge \gamma$	DS1	$\alpha \vee (\beta \wedge \gamma) \subset (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
AS2	$(\alpha \wedge \beta) \wedge \gamma \subset \alpha \wedge (\beta \wedge \gamma)$	DS2	$(\alpha \vee \beta) \wedge (\alpha \vee \gamma) \subset \alpha \vee (\beta \wedge \gamma)$
AS3	$\alpha \vee (\beta \vee \gamma) \subset (\alpha \vee \beta) \vee \gamma$	DS3	$\alpha \wedge (\beta \vee \gamma) \subset (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$
AS4	$(\alpha \vee \beta) \vee \gamma \subset \alpha \vee (\beta \vee \gamma)$	DS4	$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \subset \alpha \wedge (\beta \vee \gamma)$
COM1	$\alpha \wedge \beta \subset \beta \wedge \alpha$	AB1	$\alpha \vee (\alpha \wedge \beta) \subset \alpha$
COM2	$\alpha \vee \beta \subset \beta \vee \alpha$	AB2	$\alpha \subset \alpha \wedge (\alpha \vee \beta)$
ID1	$\alpha \subset \alpha \wedge \alpha$	ID2	$\alpha \vee \alpha \subset \alpha$
SUP	$\alpha \subset \alpha \vee \beta$	INF	$\alpha \wedge \beta \subset \alpha$
TOP	$\alpha \subset \alpha \wedge \top$	BOT	$\alpha \vee \bot \subset \alpha$
Inequality logic rules			
SUB	$\dfrac{\mathit{\Gamma} \vdash_c \alpha \subset \beta}{\mathit{\Gamma} \vdash_c \alpha\varsigma \subset \beta\varsigma}$	CTX	$\dfrac{\mathit{\Gamma} \vdash_c \alpha \subset \beta}{\mathit{\Gamma} \vdash_c \gamma[X := \alpha] \subset \gamma[X := \beta]}$
TRA	$\dfrac{\mathit{\Gamma} \vdash_c \alpha \subset \beta \quad \beta \subset \gamma}{\mathit{\Gamma} \vdash_c \alpha \subset \gamma}$	REF	$\dfrac{}{\mathit{\Gamma} \vdash_c \alpha \subset \alpha}$
Consistent consequence rules			
CC	$\dfrac{\mathit{\Gamma}, X \subset Y \vdash_c f_X \subset f_Y \qquad \mathsf{rank}(X) = \mathsf{rank}(Y)}{\mathit{\Gamma} \vdash_c X \subset Y}$		
CNT	$\dfrac{}{\mathit{\Gamma} \vdash_c X \subset Y}\ (X \subset Y) \in \mathit{\Gamma}$		

Lemma 1. *The proof system $\vdash_P$ is sound and complete for logical consequence.* □

Before we address the soundness and completeness of the proof system for consistent consequence, we parameterise the definition of $\lessdot$ to facilitate reasoning about a context $\mathit{\Gamma}$, which can be thought of as a relation on propositional variables.

Definition 7. *Let $R, \mathit{\Gamma}$ be relations on proposition variables. Let $\mathcal{E}$ be an equation system. Then R is a consistent consequence* relative to $\mathit{\Gamma}$ *if, whenever $X \ R \ X'$, for $X, X' \in \mathsf{bnd}(\mathcal{E})$, we have:*

1. $\mathsf{rank}_{\mathcal{E}}(X) = \mathsf{rank}_{\mathcal{E}}(X')$;
2. *for all $\theta \in \mathit{\Theta}_{R \cup \mathit{\Gamma}}$, we have $[\![f_X]\!]\theta \Rightarrow [\![f_{X'}]\!]\theta$.*

Proposition X' is said to be a consistent consequence of X, relative to context Γ, denoted $X \lessdot_{\Gamma} X'$ iff there is a consistent consequence relation R relative to Γ, such that $X(R \cup \Gamma)X'$.

We next generalise the notion of a consistent consequence relative to a context to apply to arbitrary proposition formulae; the generalisation is standard.

Definition 8. *Let f, g be proposition formulae. We say that g is a consistent consequence of f, relative to context Γ, denoted $f \lessdot_{\Gamma} g$ iff for all $\theta \in \Theta_{R \cup \Gamma}$, we have $[\![f]\!]\theta \Rightarrow [\![g]\!]\theta$.*

From hereon, we tacitly assume a fixed equation system $\mathcal{E}$. Henceforth, $\subset$ denotes the relation induced by all rules found in the proof system in Table 2. We write $\Gamma \vdash_c f \subset g$ to denote that the rules in Table 2 enables us to derive that $f \subset g$ in the context Γ.

Theorem 4. *The proof system $\vdash_c$ is sound for $\lessdot$.* □

Essential to our completeness result, we first claim that for all $X \lessdot Y$, we have $\vdash_c X \subset Y$; in other words, there is always a finite proof tree with root $X \subset Y$.

Lemma 2. *For all $\Gamma \subseteq \lessdot$ satisfying $f_1 \lessdot_{\Gamma} f_2$, the below rule is derivable in $\vdash_c$:*

$$\frac{H}{\Gamma \vdash_c f_1 \subset f_2} \qquad \text{(R2)}$$

where either $H = \emptyset$ or nodes in H are of the form $\Gamma' \vdash_c f \subset g$ for some γ' satisfying $\#\Gamma' > \#\Gamma$.

Theorem 5. *Proof system $\vdash_c$ is complete for $\lessdot$.*

Proof. Take any X, Y such that $X \lessdot Y$. We can construct a finite proof tree by first applying the rule R2 from Lemma 2. We thus obtain:

$$\frac{H_1}{\emptyset \vdash_c X \subset Y}$$

We continue to apply rule R2 to the premises from H_1 and to thus obtained new premises as long as necessary. On each level H_n we have $\#\Gamma = n$ and since, as a result of the finite size of an equation system, the number of possible pairs in the context Γ is finite, the proof tree is well-founded. □

Theorem 6. *The proof system conjectured in [14] is sound and complete for $\div$.* □

Example 3. As an illustration of the use of our proof system, we will prove the claimed $X_1 \lessdot X_3$ from Example 1.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{X_0 \subset X_2, X_1 \subset X_3 \vdash_c X_4 \subset X_4 \vee X_5}\,\text{(SUP)}}{X_1 \subset X_3 \vdash_c X_0 \subset X_2}\,\text{(CC)}}{X_1 \subset X_3 \vdash_c X_0 \vee X_2 \subset X_2 \vee X_2}\,\text{(CTX)} \qquad \dfrac{}{X_2 \vee X_2 \subset X_2}\,\text{(ID2)}}{X_1 \subset X_3 \vdash_c X_0 \vee X_2 \subset X_2}\,\text{(TRA)}}{X_1 \subset X_3}\,\text{(CC)}$$

6 Application

We next use the concept of consistent consequence to establish the correctness of a novel abstraction mechanism for *parameterised* Boolean equation systems. Our mechanism is inspired by abstraction techniques for behavioural systems [2].

Due to the imposed page limits, we limit ourselves to describing the syntax of parameterised Boolean equation systems; for a detailed, formal exposition of the framework, we refer to [6]. For the purpose of this section, parameterised Boolean equation systems can be thought of as describing (possibly infinite sized) blocks of Boolean equation systems with first-order right-hand side formulae. We assume some familiarity with model checking and the μ-calculus.

Definition 9. *A* parameterised Boolean equation system (PBES) $\mathcal{E}$ *is defined by the following grammar:*

$$\begin{array}{ll} \mathcal{E} & ::= \epsilon \mid (\nu X(\boldsymbol{d_X}{:}\boldsymbol{D_X}) = \phi) \mid (\mu X(\boldsymbol{d_X}{:}\boldsymbol{D_X}) = \phi) \\ \phi, \psi & ::= b \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall d{:}D.\ \phi \mid \exists d{:}D.\ \phi \mid X(\boldsymbol{e}) \end{array}$$

The expressions ϕ, ψ are predicate formulae; b is a basic Boolean expression, possibly containing data variables $d \in \mathcal{D}$. Variable X, taken from some countable set $\mathcal{P}$, is a (sorted) predicate variable; $\mathbf{e}$ is a vector of data expressions. We write $\phi(\boldsymbol{d})$ to indicate that the variables in $\mathbf{d}$ occur freely in ϕ.

PBESs allow one to encode various verification problems, such as model checking (first order) modal μ-calculus formulae and verifying various well-known process equivalences. Despite the undecidability of solving PBESs, syntactic manipulations enable one to compute the solution in many practical situations. Most manipulations preserve underlying consistent correlations. In contrast, we next present an abstraction technique that is inspired by consistent consequence.

Definition 10. *Let, for each X, $h_X{:}\boldsymbol{D_X} \to \widehat{\boldsymbol{D}}_{\boldsymbol{X}}$ be a surjection, mapping a* concrete domain $\boldsymbol{D_X}$ *to an* abstract domain $\widehat{\boldsymbol{D}}_{\boldsymbol{X}}$. *We denote the union of all mappings h_X by h. The* abstraction $\mathcal{F}^m(\phi)$ *of ϕ, for $m \in \{\sqcup, \sqcap\}$, in the context of h is defined inductively:*

$$\begin{array}{ll} \mathcal{F}^m(b(\boldsymbol{d})) & = \begin{cases} \exists \boldsymbol{d}.\ h(\boldsymbol{d}) = \widehat{\mathbf{d}} \wedge b \textit{ if } m = \sqcup \\ \forall \boldsymbol{d}.\ h(\boldsymbol{d}) \neq \widehat{\mathbf{d}} \vee b \textit{ otherwise} \end{cases} \\ \mathcal{F}^m((Y(e))(\boldsymbol{d})) & = \begin{cases} \exists \boldsymbol{d}.\ h(\boldsymbol{d}) = \widehat{\mathbf{d}} \wedge \widehat{Y}(h(e)) \textit{ if } m = \sqcup \\ \forall \boldsymbol{d}.\ h(\boldsymbol{d}) \neq \widehat{\mathbf{d}} \vee \widehat{Y}(h(e)) \textit{ otherwise} \end{cases} \\ \mathcal{F}^m((\phi \oplus \psi)(\boldsymbol{d})) & = \mathcal{F}^m(\phi(\boldsymbol{d})) \oplus \mathcal{F}^m(\psi(\boldsymbol{d})) \qquad \textit{for } \oplus \in \{\wedge, \vee\} \\ \mathcal{F}^m((\mathsf{Q}\, d'{:}D.\ \phi)(\boldsymbol{d})) & = \mathsf{Q}\, \widehat{d'}{:}\widehat{D}.\ \mathcal{F}^m(\phi(d', \boldsymbol{d})) \qquad \textit{for } \mathsf{Q} \in \{\forall, \exists\} \end{array}$$

The abstraction function $\mathcal{F}$ easily extends to PBESs. The following theorem states that $\mathcal{F}^m(\phi)$ can be used to under-approximate, resp. over-approximate the solution to a PBES. The correctness of this theorem can be established by proving the mapping h induces a consistent consequence on the predicate variables involved in a PBES; lifting the theory of consistent consequence to PBESs can be done along the lines of [14].

Theorem 7. *Let $\mathcal{E}$ be an arbitrary closed, non-empty PBES. Let h be a mapping from concrete data domains to abstract data domains. We have, for all environments η, all $X \in \mathsf{bnd}(\mathcal{E})$ and all v:*

$$[\![\mathcal{F}^{\sqcap}(\mathcal{E})]\!]\eta(\widehat{X}(h(v))) \Rightarrow [\![\mathcal{E}]\!]\eta(X(v)) \Rightarrow [\![\mathcal{F}^{\sqcup}(\mathcal{E})]\!]\eta(\widehat{X}(h(v)))$$

We illustrate the practical implications of the above theorem through an (academic) model checking problem on an infinite state space.

Example 4. Consider the infinite labelled transition system $M = \langle \mathbb{N}, \{a,b,c\}, \rightarrow \rangle$, where $\rightarrow \subseteq \mathbb{N} \times \{a,b,c\} \times \mathbb{N}$ is defined as the least set satisfying:

- for all $n \in \mathbb{N}$ satisfying $n < 10$ we have $n \xrightarrow{a} n+1$.
- for all $m, n \in \mathbb{N}$ satisfying $n \geq 10$ we have $n \xrightarrow{b} m$.
- for all $n \in \mathbb{N}$ satisfying $n \leq 5$ we have $n \xrightarrow{c} n+1$.

Note that such infinite labelled transition systems can be specified concisely using (process algebraic) languages such as mCRL2 or LOTOS. Consider the μ-calculus formula ϕ: $\nu X. \mu Y. \langle b \rangle X \vee \langle a \rangle Y$, asserting that there is an a,b-path along which action b is executed infinitely often. Following the (automated) translation described in, *e.g.* [14], we obtain the PBES $\mathcal{E}$ given below; we have, for all $v \in \mathbb{N}$, $X(v)$ is true *iff* $M, v \models \phi$.

$$\big(\nu X(n{:}\mathbb{N}) = Y(n)\big)\big(\mu Y(n{:}\mathbb{N}) = (n \geq 10 \wedge \exists m{:}\mathbb{N}. Y(m)) \vee (n < 10 \wedge X(n+1))\big)$$

Note that there is currently no technique to solve the PBES directly. Let $h{:}\mathbb{N} \rightarrow \mathbb{B}$ be defined as $h(n) = \top$ iff $n < 10$. The under-approximation $\mathcal{F}^{\sqcap}(\mathcal{E})$ is given below:

$$\begin{aligned}
(\nu\widehat{X}(\widehat{n}{:}\mathbb{B}) &= \forall n{:}\mathbb{N}.\ h(n) \neq \widehat{n} \vee \widehat{Y}(h(n)))\\
(\mu\widehat{Y}(\widehat{n}{:}\mathbb{B}) &= ((\forall n{:}\mathbb{N}.\ h(n) \neq \widehat{n} \vee n \geq 10) \wedge\\
&\qquad (\exists \widehat{m}{:}\mathbb{B}.\ (\forall n, m{:}\mathbb{N}.\ h(n) \neq \widehat{n} \vee h(m) \neq \widehat{m} \vee \widehat{Y}(h(m))))) \vee\\
&\qquad ((\forall n{:}\mathbb{N}.\ h(n) \neq \widehat{n} \vee n < 10) \wedge (\forall n{:}\mathbb{N}.\ h(n) \neq \widehat{n} \vee \widehat{X}(h(n+1)))))
\end{aligned}$$

Instantiating $\mathcal{F}^{\sqcap}(\mathcal{E})$ into a Boolean equation system yields the following result:

$$(\nu\widehat{X}_\top = \widehat{Y}_\top)\ (\nu\widehat{X}_\bot = \widehat{Y}_\bot)\ (\mu\widehat{Y}_\top = \widehat{X}_\top \wedge \widehat{X}_\bot)\ (\mu\widehat{Y}_\bot = \widehat{Y}_\top \vee \widehat{Y}_\bot)$$

The solution to this equation system is $\top$ for all equations. By the instantiation, $\widehat{X}_b = \widehat{X}(b)$ for $b \in \mathbb{B}$; by Theorem 7, we find that the solution to $\widehat{X}(h(v))$ in $\mathcal{F}^{\sqcap}(\mathcal{E})$ implies the solution to $X(v)$ in $\mathcal{E}$ for all $v \in \mathbb{N}$. By the correspondence of the solution of $\mathcal{E}$ and the problem $M, v \models \phi$, we find that $M, v \models \phi$ for every state $v \in \mathbb{N}$ of M. □

It is noteworthy that, irrespective of the problems encoded by the PBESs, our abstraction mechanism is sound; *e.g.*, it is suited for model checking problems stemming from the *full* μ-calculus. This starkly contrasts most works employing simulation relations on processes, as (unless one imposes stricter requirements on the abstractions, or moves from labelled transition systems to *modal* labelled transition systems) only the universal, resp. existential, modal μ-calculus fragments are preserved, resp. reflected [8].

To illustrate this, a slightly finer-grained abstraction function h than the one used in the above example, suffices to over-approximate the PBES resulting from the encoding of the problem $M, v \models \mu Y.\ \langle c \rangle \nu X.([a]X \wedge [b]Y)$, still giving a negative answer to the model checking problem for all states $v \in \mathbb{N}$ of M. The latter modal μ-calculus formula is neither part of the universal fragment of the μ-calculus, nor its existential fragment.

7 Future Work

The notion of a consistent consequence, together with the theory we developed in this paper, open up new avenues for research on manipulations for BESs and PBESs. We plan to investigate specialisations of the abstraction mechanisms we introduced in Section 6, such as the use of Galois connections rather than functions. Moreover, we believe it would be worthwhile to investigate the use of CEGAR techniques for automating the construction of proper abstractions.

Acknowledgements We would like to thank Bas Luttik (TU/e) and Herman Geuvers (Radboud University Nijmegen) for their useful comments and suggestions.

References

1. A. Arnold and P. Crubille. A linear algorithm to solve fixed-point equations on transition systems. *Information Processing Letters*, 20(1):57–66, 1988.
2. E.M. Clarke, O. Grumberg, and D.E. Long. Model checking and abstraction. *ACM Trans. Program. Lang. Syst.*, 16(5):1512–1542, 1994.
3. J.P. Delgrande and A. Gupta. Two results in negation-free logic. *Applied Mathematics Letters*, 6(6):79–83, 1993.
4. K. Etessami, T. Wilke, and R. A. Schuller. Fair simulation relations, parity games, and state space reduction for büchi automata. *SIAM J. Comput.*, 34(5):1159–1175, 2005.
5. C. Fritz and T. Wilke. Simulation relations for alternating parity automata and parity games. In *DLT*, volume 4036 of *LNCS*, pages 59–70. Springer, 2006.
6. J.F. Groote and T.A.C. Willemse. Parameterised boolean equation systems. *Theor. Comput. Sci*, 343(3):332–369, 2005.
7. M. Keinänen. *Techniques for Solving Boolean Equation Systems*. PhD thesis, Helsinki University of Technology, 2006.
8. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6(1):11–44, 1995.
9. A. Mader. *Verification of Modal Properties Using Boolean Equation Systems*. PhD thesis, Technische Universität München, 1997.
10. R. Mateescu. *Vérification des propriétés temporelles des programmes parallèles*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.
11. R. Mateescu. A generic on-the-fly solver for alternation-free Boolean equation systems. In *TACAS*, volume 2619 of *LNCS*, pages 81–96. Springer, 2003.
12. Robert McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65(2):149–184, 1993.
13. S. Orzan, J.W. Wesselink, and T.A.C. Willemse. Static analysis techniques for parameterised Boolean equation systems. In *TACAS*, volume 5505 of *LNCS*, pages 230–245, 2009.
14. T.A.C. Willemse. Consistent correlations for parameterised boolean equation systems with applications in correctness proofs for manipulations. In *CONCUR*, volume 6269 of *LNCS*, pages 584–598. Springer, 2010.
15. D. Zhang and R. Cleaveland. Fast generic model-checking for data-based systems. In Farn Wang, editor, *FORTE*, volume 3731 of *LNCS*, pages 83–97. Springer, 2005.