On the Notion of Vacuous Truth

Marko Samer[1] and Helmut Veith[2]

[1] Department of Computer Science
Durham University, UK
marko.samer@durham.ac.uk
[2] Institut für Informatik (I-7)
Technische Universität München, Germany
veith@in.tum.de

Abstract. The model checking community has proposed numerous definitions of vacuous satisfaction, i.e., formal criteria which tell whether a temporal logic specification holds true on a system model *for the intended reason*. In this paper we attempt to study the notion of vacuous satisfaction from first principles. We show that despite the apparently vague formulation of the vacuity problem, most proposed notions of vacuity for temporal logic can be cast into a uniform and simple framework, and compare previous approaches to vacuity detection from this unified point of view.

1 Introduction

193. *What does this mean: the truth of a proposition is certain?*
L. Wittgenstein, On Certainty [35]

Modern model checkers are equipped with capabilities which go well beyond deciding the truth of a temporal specification φ_{Spec} on a system $\mathfrak{S}$. Most importantly, when a model checker determines that the specification is violated, i.e., $\mathfrak{S} \not\models \varphi_{\mathsf{Spec}}$, it will output a counterexample, for instance a program trace, which illustrates the failure of the specification φ_{Spec} on $\mathfrak{S}$. This counterexample is a piece of evidence which the user can analyze to understand and diagnose the problem. Since counterexamples should be perceptually and mathematically simple, counterexample generation has both algorithmic and psychological aspects [12,13,17].

In this paper, we are concerned with the dual situation when the model checker asserts $\mathfrak{S} \models \varphi_{\mathsf{Spec}}$. Industrial practice shows that 20% of successful model checking passes are *vacuous*, i.e., φ_{Spec} is satisfied for some trivial or unintended reason [4]. A classical example of vacuity is *antecedent failure*, where the model checker correctly asserts

$$\mathfrak{S} \models \mathbf{AG}\,(\mathsf{trigger\_event} \Rightarrow \varphi),$$

but a closer inspection shows that in fact trigger_event is always false, and thus the implication becomes vacuously true. The total absence of trigger_event may be an indicator of erroneous system behavior, and should be reported to the user.

A model checker with automated vacuity detection thus has three kinds of outputs:

	Model Theoretic Result	**Supporting Evidence**
(i)	$\mathfrak{S} \not\models \varphi_{\mathsf{Spec}}$	Counterexample
(ii)	$\mathfrak{S} \models \varphi_{\mathsf{Spec}}$ *vacuously*	Explanation of Vacuity
(iii)	$\mathfrak{S} \models \varphi_{\mathsf{Spec}}$ *non-vacuously*	Witness of Non-Vacuity

The central question in the vacuity literature concerns the line which separates cases (ii) and (iii). In other words: When is a specification vacuously satisfied? Similar as counterexample generation, vacuity detection also relies on algorithmic and psychological insights. A recent thread of papers [1,3,4,5,6,8,11,15,18,19,22,23,25,27,28,30,33] have given different, sometimes competing, definitions, including one by the present authors [28,30].

The controversial examples and discussions of vacuity in the literature have their origin in a principal limitation of formal vacuity detection: Declaring φ_{Spec} to be vacuous on $\mathfrak{S}$ means that the specification φ_{Spec} is inadequate to capture the desired system behavior. Adequacy of specifications however is a meta-logical property that cannot be addressed inside the temporal logic, because we need *domain knowledge* to distinguish adequate specifications from inadequate ones.

The current paper develops the line of thought started in [30] in that it focuses on the notion of *vacuity grounds* as the main principle in vacuity detection. Vacuity grounds are explanations of $\mathfrak{S} \models \varphi_{\mathsf{Spec}}$, which entail the specification, but are perceptionally simpler and logically stronger. Formally, a vacuity ground is a formula φ_{Fact} such that

$$\mathfrak{S} \models \varphi_{\mathsf{Fact}} \qquad \text{and} \qquad \varphi_{\mathsf{Fact}} \models \varphi_{\mathsf{Spec}}$$

where φ_{Fact} is simpler than φ_{Spec}; criteria for simplicity will be discussed below. Thus, vacuity grounds can be viewed as a form of *interpolants* between $\mathfrak{S}$ and φ_{Spec}.

Employing vacuity grounds, it is easy to resolve conflicts between different notions of vacuity: *the same specification may be tagged as vacuous or non-vacuous, depending on which vacuity grounds the verification engineer is willing to admit*. In the antecedent failure example mentioned above, the natural vacuity ground is $\mathbf{AG}\,\neg\mathsf{trigger\_event}$. Equipped with this feedback, the verification engineer can draw a well-informed conclusion about vacuous satisfaction. We thus arrive at a revised output scheme for model checkers which support vacuity detection:

	Model Theoretic Result	**Supporting Evidence**
(i)	$\mathfrak{S} \not\models \varphi_{\mathsf{Spec}}$	Counterexample
(ii)	$\mathfrak{S} \models \varphi_{\mathsf{Spec}}$	Vacuity grounds from which the engineer decides on vacuity

We believe that our approach yields the first genuinely semantical definition of vacuity. In the rest of the paper, we compare our notion of vacuity with existing definitions, and show how our approach subsumes and uniformly explains a significant part of previous work. Moreover, we show that our approach captures cases of vacuous satisfaction not covered in the literature.

In Section 2, we review the capabilities and limitations of the prevailing definitions of vacuity – which we call unicausal semantics – and argue that they have limited explanatory power. As they are intimately tied to the syntax of φ_{Spec}, two logically equivalent specifications may become vacuous in one case and non-vacuous in the other.

Section 3 introduces our interpolation-based vacuity framework. We show that unicausal semantics yield a specific class of vacuity grounds related to uniform interpolation, thus embedding unicausal semantics into our framework. We briefly review our previously published vacuity methodology based on temporal logic query solving [30], and derive basic complexity results for the general case presented here.

Technical Preliminaries. We assume the reader is familiar with the temporal logics CTL and LTL, Kripke structures, and other standard notions. For Kripke structures $\mathfrak{S}$ and $\mathfrak{S}'$ we write $\mathfrak{S} \cong_{cnt} \mathfrak{S}'$ to denote that $\mathfrak{S}$ and $\mathfrak{S}'$ are counting-bisimilar, and we write $\mathfrak{S} \cong \mathfrak{S}'$ to denote that $\mathfrak{S}$ and $\mathfrak{S}'$ are bisimilar. We write $\varphi(\psi)$ to denote that ψ occurs once or several times as subformula in φ; we write $\varphi(x)$ to denote the formula $\varphi[\psi \leftarrow x]$ obtained from φ by replacing all occurrences of ψ by x. The formula $\varphi(x)$ is called *monotonic* in x if $\alpha \Rightarrow \beta$ implies $\varphi(\alpha) \Rightarrow \varphi(\beta)$. It holds that φ is monotonic in x if all occurrences of x are positive, and dually for anti-monotonicity. We say φ is *(semantically) unipolar* in x if φ is either monotonic or anti-monotonic in x; otherwise, we call φ multipolar. In the following, when we speak about unipolar formulas, we shall without loss of generality assume monotonicity.

2 Unicausal Vacuity Semantics

199. *The reason why the use of the expression "true or false" has something misleading about it is that it is like saying "it tallies with the facts or it doesn't", and the very thing that is in question is what "tallying" is here.* [35]

The *irrelevance of a subformula* for the satisfaction of a temporal specification φ_{Spec} is a natural indicator for vacuity: If on a model $\mathfrak{S}$, subformula ψ of φ_{Spec} can be arbitrarily modified without affecting the truth of the specification, φ_{Spec} is declared vacuous. This approach [4,5] has been the seminal paradigm for most of the research on vacuity. It has the obvious advantage that the vacuity of the specification can be (syntactically) traced back to a subformula, and that (non)-vacuity can be explained to the engineer on the grounds of the temporal logic. Syntactic vacuity however is usually hard to evaluate and not "robust" [1], i.e., dependent on the syntax of the specification and on changes in the system which are not related to the specification. The quest for efficiency and robustness has motivated new semantics for vacuity [1,8,18] which quantify over the allegedly vacuous subformulas. We shall refer to these semantics as *unicausal semantics*, since they all are attempts to obtain a (unique) formula $\forall x.\varphi(x)$ – which we call a *unicausal vacuity ground* – such that model checking $\mathfrak{S} \models \forall x.\varphi(x)$ determines the vacuity of φ on $\mathfrak{S}$ with respect to subformula ψ. The different possibilities to define the universal quantifier give rise to different unicausal semantics:

1. In the **formula semantics**, $\forall x$ ranges over all truth functions for x over the language L of $\mathfrak{S}$, i.e., $\forall x.\varphi(x)$ amounts to a (possibly infinitary) big conjunction of formulas $\bigwedge_{\theta \in L} \varphi(\theta)$.

For the following definitions, let $\star$ be a new atomic proposition which occurs neither in $\mathfrak{S}$ nor in φ. Given a structure $\mathfrak{S}$, a $\star$-labeling ℓ labels some states of $\mathfrak{S}$ with $\star$, resulting in a structure $\ell(\mathfrak{S})$.

Table 1. Evidence for non-vacuity of $\mathfrak{S} \models \varphi(\psi)$ with respect to ψ

Formula Semantics	[4,5]
A formula θ over the language of $\mathfrak{S}$ such that $\mathfrak{S} \not\models \varphi(\theta)$.	
Structure Semantics	[1]
A $\star$-labeling ℓ of $\mathfrak{S}$, such that $\ell(\mathfrak{S}) \not\models \varphi(\star)$.	
Tree Semantics	[1]
A *new* structure $\mathfrak{S}' \cong_{cnt} \mathfrak{S}$ together with a $\star$-labeling ℓ of $\mathfrak{S}'$, such that $\ell(\mathfrak{S}') \not\models \varphi(\star)$.	
Bisimulation Semantics	[18]
A *new* structure $\mathfrak{S}' \cong \mathfrak{S}$ together with a $\star$-labeling ℓ of $\mathfrak{S}'$, such that $\ell(\mathfrak{S}') \not\models \varphi(\star)$.	

2. In the **structure semantics**, $\forall x$ ranges over all labelings of $\mathfrak{S}$, i.e., $\mathfrak{S} \models \forall x.\varphi(x)$ iff for all $\star$-labelings ℓ it holds that $\ell(\mathfrak{S}) \models \varphi(\star)$.
3. In the **tree semantics**, $\forall x$ ranges over all labelings of structures counting-bisimilar to $\mathfrak{S}$, i.e., $\mathfrak{S} \models \forall x.\varphi(x)$ iff for all $\mathfrak{S}' \cong_{cnt} \mathfrak{S}$ and $\star$-labelings ℓ of $\mathfrak{S}'$ it holds that $\ell(\mathfrak{S}') \models \varphi(\star)$.
4. In the **bisimulation semantics**, $\forall x$ ranges over all labelings of structures bisimilar to $\mathfrak{S}$, i.e., $\mathfrak{S} \models \forall x.\varphi(x)$ iff for all $\mathfrak{S}' \cong \mathfrak{S}$ and $\star$-labelings ℓ of $\mathfrak{S}'$ it holds that $\ell(\mathfrak{S}') \models \varphi(\star)$.

Importantly, all four unicausal semantics coincide when $\varphi(x)$ is monotonic in x; in this case, $\forall x.\varphi(x)$ is equivalent to $\varphi(\mathsf{false})$ which can be easily model checked [22,23]. Thus, the differences between the unicausal semantics appear only when $\varphi(x)$ is multipolar with respect to x.

When comparing the different notions of unicausal vacuity, it is natural to consider the evidence that the model checker can provide in case of non-vacuity, cf. item (iii) in the first output scheme of Section 1. In the literature, this evidence was referred to as *interesting witness* [5,22,23]. Table 1 summarizes the evidence we obtain for the different unicausal semantics above.

Examples illustrating the differences between these semantics are shown in Table 2. The specification there demonstrate that even on the small structures of Figure 1, the unicausal semantics differ tremendously.

Remark 1. The structure quantifier guarantees $\forall x.\varphi(x) \models \varphi(\psi)$ only when ψ is a state formula. In case of LTL, this means that ψ is either a propositional subformula or the whole specification.

Remark 2. The quantifiers used for unicausal vacuity detection have been studied independently of vacuity. The extension of a modal logic by the *formula quantifier* remains a modal logic, because an infinitary temporal formula cannot distinguish bisimilar models [2]. The *structure quantifier* can be used to distinguish bisimulation-equivalent models, and thus, the resulting logic is not a modal logic. (For example, the formula $(\forall x.x) \vee (\forall x.\neg x)$ holds true only on a structure with a single state.) The computational

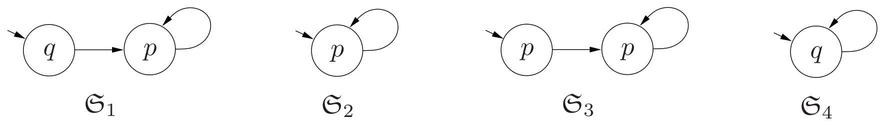

Fig. 1. Examples of Kripke structures

Table 2. Vacuity of $\varphi_1 = \mathbf{A}(p\,\mathbf{U}\,\mathbf{AG}(q \rightarrow \neg p))$, $\varphi_2 = \mathbf{AX}\,p \vee \mathbf{AX}\,\neg p$, $\varphi_3 = \mathbf{AG}\,p \vee \mathbf{AG}\,\neg p$, and $\varphi_4 = \mathbf{AG}(p \rightarrow \mathbf{AX}\neg p)$ with respect to p under different vacuity semantics. The structures are given in Figure 1 and Figure 2. Note that the non-vacuity witness $\mathfrak{S}_4$ for formula semantics is the only witness taken from Figure 1.

	Formula	**Structure**	**Tree**	**Bisimulation**
$\mathfrak{S}_1 \models \varphi_1(p)$	vacuous	vacuous	vacuous	vacuous
$\mathfrak{S}_2 \models \varphi_2(p)$	vacuous	vacuous	vacuous	$\mathfrak{S}'_0 \not\models \varphi_2(\star)$
$\mathfrak{S}_2 \models \varphi_3(p)$	vacuous	vacuous	$\mathfrak{S}'_3 \not\models \varphi_3(\star)$	$\mathfrak{S}'_3 \not\models \varphi_3(\star)$
$\mathfrak{S}_3 \models \varphi_3(p)$	vacuous	$\mathfrak{S}'_3 \not\models \varphi_3(\star)$	$\mathfrak{S}'_3 \not\models \varphi_3(\star)$	$\mathfrak{S}'_3 \not\models \varphi_3(\star)$
$\mathfrak{S}_4 \models \varphi_4(p)$	$\mathfrak{S}_4 \not\models \varphi_4(q)$	$\mathfrak{S}'_4 \not\models \varphi_4(\star)$	$\mathfrak{S}'_4 \not\models \varphi_4(\star)$	$\mathfrak{S}'_4 \not\models \varphi_4(\star)$

price to pay for the loss of modality is the undecidability of the quantified logic [16]. In combination with CTL, the *tree quantifier* is able to count the number of successors of a state [16], and thus able to break bisimulation-equivalence. While not a modal logic, the resulting logic is quite close to modal logic and retains decidability. The *bisimulation quantifier* has been rediscovered in the literature many times, and in different contexts, by the names of "bisimulation quantifier" [14], "Pitts quantifier" [34], "amorphous semantics" [16,18], and others. It is the natural quantifier to be used in the context of modal logic. Since it does not break bisimulation classes, it yields a conservative extension of a temporal logic. Uniform interpolation of the μ-calculus has been proved by elimination of bisimulation quantifiers [14].

Remark 3. Recent research has also considered vacuity detection for extensions of LTL by regular expressions [6,8]. This approach can also be viewed as an instance of unicausal semantics; for the sake of simplicity, however, we restrict the current paper to temporal logics.

2.1 Ramifications of Unicausal Vacuity

In this section, we discuss several problems and anomalies which arise from unicausal vacuity notions.

#1 Explanatory Power of Non-Vacuity Assertions
What confidence does the user gain in the model checking result when the model checker asserts non-vacuity? Recall Table 1 for the different notions of vacuity from this dual point of view:

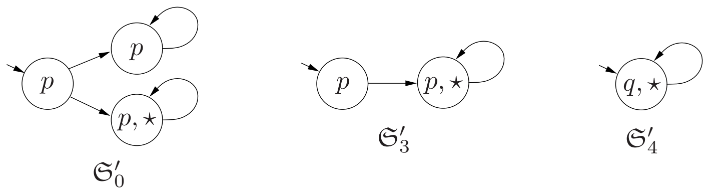

Fig. 2. Non-vacuity witnesses

- In *formula semantics*, the user knows that changing the specification $\varphi(\psi)$ into $\varphi(\theta)$ affects the truth value of the specification. Indeed, this explains the relevance of subformula ψ to the specification.
- In *structure semantics, tree semantics, and bisimulation semantics*, however, the evidence for non-vacuity is extremely weak: We change the specification $\varphi(\psi)$ into $\varphi(\star)$, where $\star$ is a new propositional symbol. Then we argue that the system $\mathfrak{S}$ (or a bisimilar system $\mathfrak{S}'$) can be labeled with $\star$ in such a way that $\varphi(\star)$ becomes false. (Thus, our non-vacuity argument is tantamount to formula semantics on a modified system with a new "imaginary" variable $\star$.) It is not clear how the introduction of $\star$ – which does not carry a meaning in the system $\mathfrak{S}$ – can give information about the relevance of subformula ψ to the user. Table 2 and its accompanying Figures 1 and 2 clearly indicate this problem.

  We see only one sound interpretation of these semantics: Suppose we know that our system $\mathfrak{S}$ is a coarse abstraction of the real system, i.e., our model $\mathfrak{S}$ is hiding many variables. Then the non-vacuity assertion states that it is consistent to assume the existence of a hidden variable $\star$ in the system which, if it were revealed, would give proof of non-vacuity in terms of formula semantics. The three semantics differ in the role of the imaginary variable $\star$: In structure semantics, variable $\star$ is uniquely defined on each state of the abstract system, while in tree semantics and bisimulation semantics, variable $\star$ depends on the execution history. A first exploration of the relationship between vacuity and abstraction was started in [18].

We conclude that only formula semantics gives confidence in the non-vacuity assertion. The other semantics provide tangible non-vacuity evidence only in very specific circumstances.

#2 Explanatory Power of Vacuity Assertions

What conclusions can the user draw from an assertion of vacuity? As temporal logic does not have quantifier elimination (cf. Section 3), it is in many cases impossible to write the vacuity ground $\forall x.\varphi(x)$ in plain temporal logic. The unicausal semantics, however, have the useful feature that each assertion of vacuity actually points out one or several subformulas which cause vacuous satisfaction. Comparing the different unicausal semantics, we obtain the following picture:

- In formula semantics, the vacuity assertion is relatively easy to understand: it says that no syntactic change in the subformulas of interest causes the specification to fail.

- In the other semantics, we are facing a problem dual to #1: The vacuity assertion expresses the fact that no hidden imaginary variable $\star$ can make $\varphi(\star)$ false. This criterion is stronger than the intuitive notion of vacuity – i.e., it will detect vacuity only in few cases.

We conclude that formula semantics again yields the most natural notion of unicausal vacuity, while the other three semantics report vacuity too rarely. This is dual to our observation in #1 that the semantics give weak evidence of non-vacuity.

#3 Expressive Power of Subformula Quantification
Vacuity detection by quantification over subformula occurrences entails a number of limitations discussed below.

#3.1 **Pnueli's Observation**
Pnueli [26] pointed out that a satisfied specification $\mathbf{AG}\,\mathbf{AF}\,p$ may be considered vacuous, when the model checker observes that the stronger formula $\mathbf{AG}\,p$ is true on the system. None of the unicausal vacuity notions is able to detect this notion of vacuity.

#3.2 **Syntactically Unrelated Observations**
Following Pnueli's example, there are arguable cases of vacuity where no syntactic relationship exists between the specification and the observation. For example, for a specification $\mathbf{EF}\,p$, the model checker may observe that in fact $\mathbf{AX}\,p$ holds. It is clear that this form of vacuity cannot be detected by unicausal semantics.

#3.3 **Specifications with Single Propositions**
The issues raised in #3.1 and #3.2 share the syntactic property that they contain a single occurrence of a propositional variable p. Consequently, the universal quantifier eliminates the truth-functional dependence on the propositional variable, and the quantified specification is either a tautology or a contradiction. For example, in Pnueli's example, quantification yields the formulas $\mathbf{AG}\,\mathsf{false}$ and $\mathbf{AG}\,\mathbf{AF}\,\mathsf{false}$ both of which are equivalent to false.[1] This proves that the observations (vacuity grounds) such as $\mathbf{AG}\,p$ from #3.1 and $\mathbf{AX}\,p$ from #3.2 are impossible in unicausal semantics.

#3.4 **Vacuity by Disjunction**
In the syntactic view of unicausal vacuity, certain specifications are always vacuous. In particular, if a disjunctive specification $\varphi \vee \psi$ holds true in a structure, then either $\varphi \vee \mathsf{false}$ or $\mathsf{false} \vee \psi$ holds true, and thus, the specification is inevitably vacuous.

#3.5 **Stability Under Logical Equivalence**
As explained in #3.4 above, $\varphi \vee \varphi$ is vacuous by construction, and thus, every specification is equivalent to a vacuous specification. A more interesting example is given by $\mathbf{EF}\,p$ which is equivalent to $p \vee \mathbf{EX}\,\mathbf{EF}\,p$. In the second formulation, the specification is always vacuous.

[1] Recall that in the unipolar case, universal quantification over a variable x is tantamount to setting it false.

Getting back to Pnueli's problem in #3.1, we even see that a reformulation of the specification $\mathbf{AG}\,\mathbf{AF}\,p$ into the equivalent specification $\mathbf{AG}(p \vee \mathbf{AF}\,p)$ enables us to quantify out the subformula $\mathbf{AF}\,p$, yielding a formula $\forall x.\mathbf{AG}(p \vee x)$ which is equivalent to $\mathbf{AG}(p \vee \mathsf{false})$, and thus to $\mathbf{AG}\,p$. Similarly, the problem raised in #3.3 can be solved using the specification $(\mathbf{EF}\,p) \vee (\mathbf{AX}\,p)$ instead of the (logically equivalent) specification $\mathbf{EF}\,p$.

#4 Causality
Our final concern (and indeed the original motivation for this research) is a fundamental issue raised by unicausal semantics. Given a specification $\varphi(\psi)$ and occurrences of a subformula ψ, each unicausal semantics associates the vacuity of φ with respect to ψ with a uniquely defined formula $\forall x.\varphi(x)$. Not only does such a construction revert the natural order between cause ($\mathfrak{S} \models \forall x.\varphi(x)$) and effect ($\mathfrak{S} \models \varphi(\psi)$), it is also independent of the system $\mathfrak{S}$. Thus, unicausal semantics represents a fairly simple instance of logical abduction.

3 Interpolation-Based Vacuity Detection

200. *Really "The proposition is either true or false" only means that it must be possible to decide for or against it. But this does not say what the ground for such a decision is like.* [35]

Recall from Section 1 that we define a vacuity ground as a simple formula φ_{Fact} such that

$$\mathfrak{S} \models \varphi_{\mathsf{Fact}} \qquad \text{and} \qquad \varphi_{\mathsf{Fact}} \models \varphi_{\mathsf{Spec}}. \tag{1}$$

Vacuity grounds serve as feedback for the verification engineer which helps him/her to decide whether the specification φ_{Spec} is vacuously satisfied. For example, in the cases #3.1, #3.2, and #3.4 above, natural candidates for vacuity grounds are $\mathbf{AG}\,p$, $\mathbf{AX}\,p$, and φ, respectively. In general, a model checker may output multiple vacuity grounds for a single specification.

When the vacuity grounds are chosen among modal temporal formulas, definition (1) is equivalent to

$$\chi_{\mathfrak{S}} \models \varphi_{\mathsf{Fact}} \qquad \text{and} \qquad \varphi_{\mathsf{Fact}} \models \varphi_{\mathsf{Spec}}. \tag{2}$$

where $\chi_{\mathfrak{S}}$ is the temporal formula which characterizes $\mathfrak{S}$ up to bisimulation equivalence. Thus, the vacuity ground φ_{Fact} is an *interpolant* between the system description $\chi_{\mathfrak{S}}$ and the specification φ_{Spec}. Consequently, vacuity analysis can be viewed as the process of finding *simple* interpolants between the system and the specification. Note that, technically, φ_{Spec} is usually itself a Craig interpolant, but not useful in vacuity detection. Thus, we need different notions of simplicity than the restriction to common variables.

3.1 Unicausal Vacuity Grounds and Interpolation

The unicausal vacuity grounds of Section 2 represent a specific *construction principle* for vacuity grounds using universal quantification, i.e., we have

$$\mathfrak{S} \models \forall x.\varphi(x) \qquad \text{and} \qquad \forall x.\varphi(x) \models \varphi(\psi) \tag{3}$$

with the special case

$$\mathfrak{S} \models \varphi(\mathsf{false}) \qquad \text{and} \qquad \varphi(\mathsf{false}) \models \varphi(\psi) \tag{4}$$

when $\varphi(\psi)$ is unipolar. Since the implication $\forall x.\varphi(x) \models \varphi(\psi)$ is a consequence of the construction, a model checking result $\mathfrak{S} \models \forall x.\varphi(x)$ indeed says that $\forall x.\varphi(x)$ is *one* vacuity ground. Consequently, with the exception of the special case mentioned in Remark 1, unicausal vacuity semantics indeed has a natural embedding into our framework.

The construction principle for unicausal vacuity grounds is itself closely related to interpolation. One of the main logical motivations for the introduction of propositional temporal quantifiers are proofs of Craig interpolation. In particular, uniform Craig interpolation of the μ-calculus was shown by quantifier elimination of bisimulation quantifiers [14]: Given a formula α and β where $\alpha \models \beta$, an interpolant is obtained by the quantified formula $\forall \mathbf{x}.\beta$, where $\mathbf{x}$ is the tuple of variables occurring only in β. By construction, it holds that

$$\alpha \models \forall \mathbf{x}.\beta \qquad \text{and} \qquad \forall \mathbf{x}.\beta \models \beta \tag{5}$$

whence it is sufficient to show that $\forall \mathbf{x}.\beta$ is equivalent to a quantifier-free formula to prove interpolation for the μ-calculus. Since this construction depends only on β and $\mathbf{x}$, $\forall \mathbf{x}.\beta$ is called a post-interpolant or right interpolant. (Existential quantification of α naturally yields pre-interpolants or left interpolants.)

CTL and LTL are well known not to have interpolation [24] because CTL and LTL do not admit elimination of bisimulation quantifiers. The quantified extensions of CTL and LTL, however, do have uniform interpolation, and the post-interpolants are naturally obtained by universal quantification analogously to (5). *Consequently, we conclude that unicausal vacuity grounds are generalizations of post-interpolants: they are the weakest formulas which imply φ_{Spec} without mentioning certain variables or subformulas that occur in φ_{Spec}.*

Let us note that our adversarial discussion of unicausal semantics in Section 2 does not inhibit the use of $\forall x.\varphi(x)$ *as* vacuity ground in the sense described here. The discussion of Section 2 only shows that no unicausal semantics by itself can adequately solve the problem of vacuity detection.

3.2 Computation of Vacuity Grounds

The discussion of Section 3.1 shows that the unicausal semantics yield a natural class of vacuity grounds. In the important unipolar case (cf. condition (4)), the vacuity grounds $\varphi(\mathsf{false})$ satisfy all important criteria: they are simpler than the specification, they are easy to model check, and they represent tangible feedback for the verification engineer.

When the quantifier in $\forall x.\varphi(x)$ cannot be eliminated, the situation is more complicated. As discussed in Section 2 (#2), the semantics of quantified temporal formulas has only limited explanatory value concerning vacuity. Moreover, the examples in Table 2 demonstrate that the verification engineer may obtain different vacuity feedback from

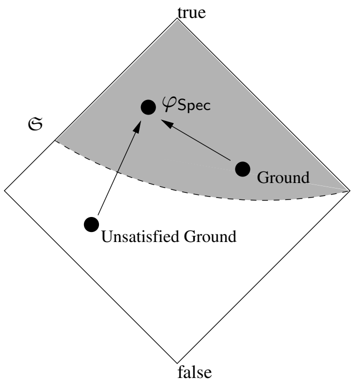

Fig. 3. In the lattice of temporal properties, system $\mathfrak{S}$ partitions the properties into satisfied properties (shaded dark) and unsatisfied ones. Vacuity grounds are located lower in the lattice order than the specification, but inside the shaded area of satisfied properties. The closer a vacuity ground is to the white area, the stronger is the intuitive strength of the vacuity assertion. The figure shows one satisfied vacuity ground and one unsatisfied ground.

different unicausal semantics, i.e., the vacuity ground for one semantics may hold true, while it is false for the other semantics. To appreciate this situation, the engineer has to understand the subtleties of the different semantics.

As argued in Section 2, it is important to obtain vacuity grounds different from the unicausal grounds. The search space for these vacuity grounds is illustrated in Figure 3. Not surprisingly, finding small vacuity grounds has the same complexity as the respective decision problem for validity: Let VAC-CTL be the decision problem if for a given structure $\mathfrak{S}$, a CTL specification φ_{Spec}, and an integer $k < |\varphi_{\mathsf{Spec}}|$, there exists a vacuity ground φ_{Fact} such that $|\varphi_{\mathsf{Fact}}| \leq k$ and $\varphi_{\mathsf{Spec}} \not\equiv \varphi_{\mathsf{Fact}}$. VAC-LTL is defined analogously for LTL. Then the following theorem is not hard to show:

Theorem 1. VAC-CTL *is* EXPTIME-*complete and* VAC-LTL *is* PSPACE-*complete.*

Thus, the complexity is not worse than checking $\varphi_{\mathsf{Fact}} \models \varphi_{\mathsf{Spec}}$. Nevertheless, it is a reasonable strategy to focus on methods which systematically enumerate perceptionally simple candidates for vacuity grounds; these methods may be based both on heuristics and formal considerations. As in the unicausal semantics, candidates φ_{Fact} will typically be chosen in such a way that $\varphi_{\mathsf{Fact}} \models \varphi_{\mathsf{Spec}}$ follows by construction, and $\mathfrak{S} \models \varphi_{\mathsf{Fact}}$ is determined by the model checker, cf. condition (1).

To obtain candidate formulas without expensive validity checks, one can systematically compute the closure of the specification under two syntactic operations, namely by rewriting and strengthening:

(i) Replace subformulas of the specification by equivalent subformulas typically involving disjunction, e.g., $\mathbf{EF}\, p$ by $p \vee \mathbf{EX}\,\mathbf{EF}\, p$, or $\mathbf{AF}\, p$ by $p \vee \mathbf{AX}\,\mathbf{AF}\, p$, etc.[2]

[2] Recent work by the authors [28,29,31] has characterized the distributivity over conjunction of temporal operators which yields also an analogous characterizations for disjunction. Such characterizations can be used to support rewriting of specifications.

(ii) Replace subformulas by stronger non-equivalent subformulas, e.g., replace whole subformulas by false as in unicausal semantics, $\mathbf{AF}\,p$ by p as in Pnueli's example, $\mathbf{EF}\,p$ by $\mathbf{AX}\,p$, $\mathbf{AX}\,p \vee \mathbf{AX}\,\neg p$ by $\mathbf{AG}\,p$, etc.

While this heuristic enumeration of antecedents naturally samples the space of possible vacuity grounds for φ_{Spec}, each candidate has to be model checked separately, similar as in unicausal semantics.

An alternative systematic approach for finding antecedents by strengthening subformulas in the unipolar case was presented in a predecessor paper [30] which introduced *parameterized vacuity*, a new approach to vacuity using temporal logic query solving. A temporal logic query solver [9] is a variant of a model checker which on input of a formula $\varphi(x)$ and a model $\mathfrak{S}$ finds the strongest formulas ψ such that $\mathfrak{S} \models \varphi(\psi)$. Such a formula ψ is called a *solution* of $\varphi(x)$ in $\mathfrak{S}$. To use temporal logic queries for computing vacuity grounds, we are interested in solutions ψ such that $\varphi(\psi)$ is a vacuity ground. In this way we are able to reduce vacuity detection to temporal logic query solving [30]. Our approach was motivated by the failure of unicausal semantics to handle Pnueli's problem. Recall that unicausal semantics cannot find vacuity ground $\mathbf{AG}\,p$ for $\mathbf{AF}\,\mathbf{AG}\,p$, cf. Section 2 (#3.1). Instead of reducing $\varphi(\psi)$ to a unicausal ground $\varphi(\mathsf{false})$, we use a temporal logic query solver to find a simple formula θ which implies ψ. Then, by monotonicity, it follows that $\varphi(\theta) \models \varphi(\psi)$. Thus, we obtain a vacuity ground $\varphi(\theta)$ which explains $\varphi(\psi)$ and solves Pnueli's problem. Several algorithms for solving temporal logic queries have been proposed in the literature; in particular, symbolic algorithms [9,28,32], automata-theoretic algorithms [7], and algorithms based on multi-valued model checking [10,20,21].

4 Conclusion

We have argued that vacuity of temporal specifications cannot be adequately captured by formal criteria. As vacuity expresses the inadequacy of a specification, it needs to be addressed by the verification engineer. We have therefore proposed a new approach to vacuity where the model checker itself does not decide on vacuity, but computes an interpolant – called a vacuity ground – which expresses a simple reason that renders the specification true. Candidate formulas for the interpolants can be obtained from existing notions of unicausal vacuity, from heuristics, and from temporal logic query solving. Equipped with the feedback vacuity grounds, the verification engineer can decide if the specification is vacuously satisfied.

The current paper has focused on the logical nature of vacuous satisfaction rather than on practical vacuity detection algorithms. We believe that future work should address the systematic computation of vacuity grounds, because vacuity is in the eye of the beholder.

Acknowledgments. The authors are grateful to Arie Gurfinkel, Kedar Namjoshi, and Richard Zach for discussions on vacuity.

References

1. Armoni, R., Fix, L., Flaisher, A., Grumberg, O., Piterman, N., Tiemeyer, A., Vardi, M.Y.: Enhanced vacuity detection in linear temporal logic. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 368–380. Springer, Heidelberg (2003)
2. Barwise, J., van Benthem, J.: Interpolation, preservation, and pebble games. Journal of Symbolic Logic 64(2), 881–903 (1999)
3. Beatty, D.L., Bryant, R.E.: Formally verifying a microprocessor using a simulation methodology. In: DAC 1994. Proc. 31st Annual ACM IEEE Design Automation Conference, pp. 596–602. ACM Press, New York (1994)
4. Beer, I., Ben-David, S., Eisner, C., Rodeh, Y.: Efficient detection of vacuity in ACTL formulas. In: Grumberg, O. (ed.) CAV 1997. LNCS, vol. 1254, pp. 279–290. Springer, Heidelberg (1997)
5. Beer, I., Ben-David, S., Eisner, C., Rodeh, Y.: Efficient detection of vacuity in temporal model checking. Formal Methods in System Design (FMSD) 18(2), 141–163 (2001)
6. Ben-David, S., Fisman, D., Ruah, S.: Temporal antecedent failure: Refining vacuity. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 492–506. Springer, Heidelberg (2007)
7. Bruns, G., Godefroid, P.: Temporal logic query checking. In: LICS 2001. Proc. 16th Annual IEEE Symposium on Logic in Computer Science, pp. 409–417. IEEE Computer Society Press, Los Alamitos (2001)
8. Bustan, D., Flaisher, A., Grumberg, O., Kupferman, O., Vardi, M.Y.: Regular vacuity. In: Borrione, D., Paul, W. (eds.) CHARME 2005. LNCS, vol. 3725, pp. 191–206. Springer, Heidelberg (2005)
9. Chan, W.: Temporal-logic queries. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, pp. 450–463. Springer, Heidelberg (2000)
10. Chechik, M., Gurfinkel, A.: TLQSolver: A temporal logic query checker. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 210–214. Springer, Heidelberg (2003)
11. Chockler, H., Strichman, O.: Easier and more informative vacuity checks. In: MEMOCODE 2007, pp. 189–198. IEEE Computer Society Press, Los Alamitos (2007)
12. Clarke, E.M., Jha, S., Lu, Y., Veith, H.: Tree-like counterexamples in model checking. In: LICS 2002. Proc. 17th Annual IEEE Symposium on Logic in Computer Science, pp. 19–29. IEEE Computer Society Press, Los Alamitos (2002)
13. Clarke, E.M., Veith, H.: Counterexamples revisited: Principles, algorithms, applications. In: Dershowitz, N. (ed.) Verification: Theory and Practice. LNCS, vol. 2772, pp. 208–224. Springer, Heidelberg (2004)
14. D'Agostino, G., Hollenberg, M.: Logical questions concerning the μ-calculus: Interpolation, Lyndon and Loś-Tarski. Journal of Symbolic Logic 65(1), 310–332 (2000)
15. Dong, Y., Sarna-Starosta, B., Ramakrishnan, C., Smolka, S.A.: Vacuity checking in the modal mu-calculus. In: Kirchner, H., Ringeissen, C. (eds.) AMAST 2002. LNCS, vol. 2422, Springer, Heidelberg (2002)
16. French, T.: Decidability of quantified propositional branching time logics. In: Stumptner, M., Corbett, D.R., Brooks, M. (eds.) AI 2001: Advances in Artificial Intelligence. LNCS (LNAI), vol. 2256, pp. 165–176. Springer, Heidelberg (2001)
17. Groce, A., Kroening, D.: Making the most of BMC counterexamples. Electronic Notes in Theoretical Computer Science (ENTCS) 119(2), 67–81 (2005)
18. Gurfinkel, A., Chechik, M.: Extending extended vacuity. In: Hu, A.J., Martin, A.K. (eds.) FMCAD 2004. LNCS, vol. 3312, pp. 306–321. Springer, Heidelberg (2004)
19. Gurfinkel, A., Chechik, M.: How vacuous is vacuous? In: Jensen, K., Podelski, A. (eds.) TACAS 2004. LNCS, vol. 2988, pp. 451–466. Springer, Heidelberg (2004)

20. Gurfinkel, A., Chechik, M., Devereux, B.: Temporal logic query checking: A tool for model exploration. IEEE Transactions on Software Engineering (TSE) 29(10), 898–914 (2003)
21. Gurfinkel, A., Devereux, B., Chechik, M.: Model exploration with temporal logic query checking. In: Proc. 10th International Symposium on the Foundations of Software Engineering (FSE-10), pp. 139–148. ACM Press, New York (2002)
22. Kupferman, O., Vardi, M.Y.: Vacuity detection in temporal model checking. In: Pierre, L., Kropf, T. (eds.) CHARME 1999. LNCS, vol. 1703, pp. 82–96. Springer, Heidelberg (1999)
23. Kupferman, O., Vardi, M.Y.: Vacuity detection in temporal model checking. International Journal on Software Tools for Technology Transfer (STTT) 4(2), 224–233 (2003)
24. Maksimova, L.: Absence of interpolation and of Beth's property in temporal logics with "the next" operation. Siberian Mathematical Journal 32(6), 109–113 (1991)
25. Namjoshi, K.S.: An efficiently checkable, proof-based formulation of vacuity in model checking. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 57–69. Springer, Heidelberg (2004)
26. Pnueli, A.: 9th International Conference on Computer-Aided Verification. In: Grumberg, O. (ed.) CAV 1997. LNCS, vol. 1254, Springer, Heidelberg (1997) (cited from [5])
27. Purandare, M., Somenzi, F.: Vacuum cleaning CTL formulae. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 485–499. Springer, Heidelberg (2002)
28. Samer, M.: Reasoning about Specifications in Model Checking. PhD thesis, Vienna University of Technology (2004)
29. Samer, M., Veith, H.: Validity of CTL queries revisited. In: Baaz, M., Makowsky, J.A. (eds.) CSL 2003. LNCS, vol. 2803, pp. 470–483. Springer, Heidelberg (2003)
30. Samer, M., Veith, H.: Parameterized vacuity. In: Hu, A.J., Martin, A.K. (eds.) FMCAD 2004. LNCS, vol. 3312, pp. 322–336. Springer, Heidelberg (2004)
31. Samer, M., Veith, H.: A syntactic characterization of distributive LTL queries. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 1099–1110. Springer, Heidelberg (2004)
32. Samer, M., Veith, H.: Deterministic CTL query solving. In: TIME 2005, pp. 156–165. IEEE Computer Society Press, Los Alamitos (2005)
33. Simmonds, J., Davies, J., Gurfinkel, A., Chechik, M.: Exploiting resolution proofs to speed up LTL vacuity detection for BMC. In: FMCAD 2007. Proc. 7th International Conference on Formal Methods in Computer-Aided Design, IEEE Computer Society Press, Los Alamitos (to appear, 2007)
34. Visser, A.: Bisimulations, model descriptions and propositional quantifiers. Logic Group Preprint Series, Nbr. 161, Dept. Philosophy, Utrecht University (1996)
35. Wittgenstein, L.: On Certainty. In: Anscombe, G.E.M., von Wright, G.H. (eds.) Harper and Row (1968)