How Thorough is Thorough Enough?

Arie Gurfinkel and Marsha Chechik

Department of Computer Science, University of Toronto,
Email: {arie,chechik}@cs.toronto.edu

Abstract. Abstraction is the key for effectively dealing with the state explosion problem in model-checking. Unfortunately, finding abstractions which are small and yet enable us to get conclusive answers about properties of interest is notoriously hard. Counterexample-guided abstraction refinement frameworks have been proposed to help build good abstractions iteratively. Although effective in many cases, such frameworks can include unnecessary refinement steps, leading to larger models, because the abstract verification step is not as conclusive as it can be in theory. Abstract verification can be supplemented by a more precise but much more expensive *thorough* check, but it is not clear how often this check really helps. In this paper, we study the relationship between model-checking and thorough checking and identify practical cases where the latter is not necessary, and those where it can be performed efficiently.

1 Introduction

Abstraction is arguably the most effective technique for dealing with the state explosion problem in model-checking. The goal of abstraction is to build a system which is small enough to analyze yet the one that allows to verify properties of interest. Such abstractions may be very hard to build; instead, we typically start with an abstraction which may be too crude for certain properties, and then refine it, attempting to reach a definite answer.

The best-known method for abstraction refinement, guided by counterexamples, has been suggested by Clarke et al. [5] and is outlined in Figure 1(a). This framework assumes that the abstraction K_α is an overapproximation of the system of interest K_c, i.e., every execution of K_c is an execution of K_α. When a universal property φ holds in K_α, this result can be trusted. Otherwise, either φ does not hold in K_c, or the abstraction is too crude. To tell between these cases, a counterexample obtained by verifying φ in K_α is checked for feasibility by playing it back in K_c. This either establishes the failure of φ, or enables the refinement of K_α that eliminates the spurious counterexample.

Several researchers [12, 22, 4, 9, 11] proposed an improvement of this framework that enables reasoning about arbitrary CTL formulas. In their framework, outlined in Figure 1(b), an abstract model K_α is 3-valued, which combines over- and under-approximation of K_c. Model-checking a CTL formula φ on K_α either yields true or false, which can be trusted without the need to resort to the counterexample, or it returns maybe, i.e., inconclusive. In this case, the counterexample can be used to refine the abstraction. Since building 3-valued models is no more expensive than classical [11], and neither is 3-valued model-checking [4] nor 3-valued counterexample generation [15, 22], this framework is not more expensive than classical, while allowing to reason about a larger class of temporal logic properties.

Goal: Check ACTL formula φ on a model K_c

1. **repeat** until resources are exhausted
2. Build an abstract model K_α.
3. Model-check φ on K_α.
4. **if** YES, **return** "φ holds on K_c"
5. **else**
6. Check if the counterexample is feasible
7. **if** YES, **return** "φ fails on K_c"
8. **else** use the counterexample for refinement.

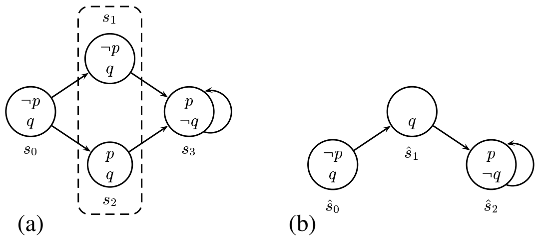
(a)

Goal: Check CTL formula φ on a model K_c

1. **repeat** until resources are exhausted
2. Build a 3-val abstract model K_α.
3. Model-check φ on K_α.
4. **if** YES, **return** "φ holds on K_c"
5. **if** NO, **return** "φ fails on K_c"
6. **else** use the counterexample for refinement.

(b)

Fig. 1. Counterexample-guided abstraction refinement frameworks: (a) classical; (b) 3-valued..



(a) (b)

Fig. 2. (a) A concrete model K; (b) An abstraction K' of K.

Both of these frameworks sometimes force a refinement step even though a conclusive result can be obtained from the existing model K_α. For example, consider checking a property $\varphi = A[(\neg p \wedge q)\ U\ p]$, where the original model K and its abstraction K' are shown in Figure 2. In K', states $\hat{s}_0$ and $\hat{s}_2$ correspond to s_0 and s_3 of K, respectively, whereas $\hat{s}_1$ is a merge of s_1 and s_2, indicated by dashed lines in Figure 2(a). In classical abstraction, we typically treat literals of the concrete models as atomic propositions of the abstract, thus both p and $\neg p$ are false in state $\hat{s}_1$ of K'. Our property φ fails in K', and a counterexample is produced. Clearly, this counterexample is not feasible, so refinement is necessary. On a closer inspection, we note that this counterexample is spurious not only in K but in *every* model that refines K'. There are two concretizations of this counterexample, and φ is true in both of them. Thus, it would be highly desirable to be able to conclude that the property holds, avoiding unnecessary refinement steps.

Godefroid and Jagadeesan [12] suggested that one can use an additional, *thorough*, check when the result of model-checking is inconclusive. This changes both algorithms in Figure 1 after step 5 as follows:

5a. Apply the thorough check of φ on K_α.
5b. **if** conclusive, tell user and stop.
6. **else** use the counterexample for refinement.

which we refer to as *classical thorough* and *3-valued thorough*, respectively. Even though the thorough check is exponentially more expensive than model-checking [12], this modification can potentially reduce the number of refinements. Since each refinement can lead to an exponential increase in the size of the abstraction, the extra cost seems justified. Unfortunately, if the thorough check is still inconclusive, it does not help the refinement, but levies a heavy performance penalty. Without empirical evidence, it is not clear how useful this framework is in practice. We are thus interested to find out answers to the following questions:

1. Are there classes of problems where the thorough check is not necessary, i.e., it does not give a more precise result than model-checking?

2. In cases where the thorough check is required, can it be performed efficiently?

In this paper, we show that the thorough check of universal properties on models built using *predicate abstraction* [14] does not give an additional precision and thus can be skipped. For arbitrary abstraction, we give an algorithm for deciding ACTL formulas, where the thorough check can be performed efficiently. This approach combines the model-checking and the thorough step, resulting in an algorithm which is as precise as the thorough check, while being only marginally more expensive than model-checking. This algorithm also produces counterexamples which can be used for refinement.

The rest of this paper is organized as follows. We start by giving the necessary background in Section 2. In Section 3, we extend results of Godefroid and Jagadeesan [13] to show that 3-valued models in which each atomic proposition is either boolean (i.e., true or false), or maybe in each state, are as expressive as arbitrary 3-valued Kripke structures. This is used in Section 4 to show that 3-valued model-checking (referred to as *compositional*) and thorough checking correspond to different semantics of quantified temporal logic (QTL). We answer the questions posed above in Section 5, using previously established results for QTL. We compare our approach with related work in Section 6 and conclude the paper in Section 7.

2 Background

In this section, we provide the necessary background on model-checking, 3-valued reasoning, and quantified temporal logic.

3-Valued Kleene Logic. A 3-valued Kleene logic [18] is an extension of a classical two-valued logic of true and false, with an additional value maybe, representing uncertainty. Logical operators in the logic are defined via the *truth* ordering $\sqsubseteq$, where false $\sqsubseteq$ maybe $\sqsubseteq$ true. Intuitively, $a \sqsubseteq b$ indicates that a is *less true* than b. Conjunction and disjunction are given by meet (minimum) and join (maximum) operators of the truth ordering, respectively. Negation is defined as: $\neg$true $=$ false, $\neg$false $=$ true, and $\neg$maybe $=$ maybe. Kleene logic preserves most of the laws of classical logic, such as De Morgan laws ($\neg(a \wedge b) = \neg a \vee \neg b$), and an involution of negation ($\neg\neg a = a$), but not the laws of excluded middle ($a \vee \neg a =$ true) and non-contradiction ($\neg a \wedge a =$ false). The values of Kleene logic can also be ordered according to the *information* pre-order $\preceq$, where maybe $\preceq$ true and maybe $\preceq$ false. That is, maybe contains the least amount of information, whereas true and false are incomparable. We denote the set of boolean values true and false by $\mathbf{2}$, and the set of values of Kleene logic by $\mathbf{3}$.

Models. A model is a 3-valued Kripke structure $K = (S, R, S_0, AP, I)$, where S is a finite set of states, $R : S \times S \to \mathbf{3}$ is a total transition relation, $S_0 \subseteq S$ is a set of initial states, AP is a set of atomic propositions, and $I : S \times AP \to \mathbf{3}$ is an interpretation function, assigning a value to each atomic proposition $a \in AP$ in each state. A classical (two-valued) Kripke structure is a 3-valued Kripke structure that does not use the value maybe, i.e. the range of R and I is $\{\text{true}, \text{false}\}$.

Temporal Logic. *Computation Tree Logic* (CTL) [7] is a branching temporal logic, whose syntax is defined with respect to set AP of atomic propositions, as follows:

$$\begin{aligned}\varphi = \ell \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid EX\varphi \mid AX\varphi \mid EF\varphi \mid AF\varphi \\ \mid EG\varphi \mid AG\varphi \mid E[\varphi\, U\, \varphi] \mid A[\varphi\, U\, \varphi],\end{aligned}$$

$$
\begin{array}{rlrl}
||\ell||^K(s) & \triangleq \ell & ||p||^K(s) & \triangleq I(s,p) \\
||\varphi \wedge \psi||^K(s) & \triangleq ||\varphi||^K(s) \wedge ||\psi||^K(s) & ||\varphi \vee \psi||^K(s) & \triangleq ||\varphi||^K(s) \vee ||\psi||^K(s) \\
||\neg\varphi||^K(s) & \triangleq \neg||\varphi||^K(s) & ||EX\varphi||^K(s) & \triangleq \bigvee_{t\in S}(R(s,t) \wedge ||\varphi||^K(t)) \\
||EG\varphi||^K(s) & \triangleq ||\nu Z \cdot \varphi \wedge EXZ||^K(s) & ||E[\varphi U \psi]||^K(s) & \triangleq ||\mu Z \cdot \psi \vee \varphi \wedge EXZ||^K(s)
\end{array}
$$

Fig. 3. Semantics of CTL.

where $p \in AP$ is an atomic proposition and $\ell \in \mathbf{2}$ is a constant. Informally, the meaning of the temporal operators is: given a state and all paths emanating from it, φ holds in one (EX) or all (AX) next states; φ holds in some future state along one (EF) or all (AF) paths; φ holds globally along one (EG) or all (AG) paths, and φ holds until a point where ψ holds along one (EU) or all (AU) paths.

The value of φ in state s of K is denoted by $||\varphi||^K(s)$; the value of φ in K is defined with respect to all initial states of K: $||\varphi||^K = \bigwedge_{s_0 \in S_0} ||\varphi||^K(s_0)$. Temporal operators EX, EG, and EU together with the propositional connectives form an adequate set [6]. The formal semantics of CTL is given in Figure 3. The only difference between the 2- and the 3-valued semantics is the change in the domain of ℓ. To disambiguate from an alternative semantics presented below, we refer to this semantics as *compositional*. Compositional semantics of CTL is interpreted over 3-valued Kripke structures with respect to Kleene logic.

We write $\varphi[x]$ to indicate that the formula φ may contain an atomic proposition x. An occurrence of x is *positive* (or of *positive polarity*) if it occurs under the scope of an even number of negations, and *negative* otherwise. An atomic proposition x is *pure* in φ if all of its occurrences have the same polarity, and is *mixed* otherwise. We write $\varphi[x \leftarrow y]$ for a formula obtained from φ by simultaneously substituting all occurrences of x by y. A formula φ is *universal* (or in ACTL) if all of its temporal operators are universal, and is *existential* (or in ECTL) if they are existential. In both cases, negation is only allowed at the level of atomic propositions.

Relationships Between Models. We revisit definitions of simulation and bisimulation for classical Kripke structures, and refinement for 3-valued Kripke structures.

Definition 1. *[20] Let K and K' be classical Kripke structures with identical sets of atomic propositions AP. A relation $\rho \subseteq S \times S'$ is a* simulation *iff $\rho(s, s')$ implies that*

1. $\forall p \in AP \cdot I'(s', p) \Leftrightarrow I(s, p)$*, and*
2. $\forall t' \in S' \cdot R'(s', t') \Rightarrow \exists t \in S \cdot R(s, t) \wedge \rho(t, t')$.

A state s *simulates* a state s' if $(s, s') \in \rho$. A Kripke structure K simulates K' iff every initial state of K' is simulated by an initial state of K. Simulation between K and K' preserves ACTL: for any $\varphi \in$ ACTL, $||\varphi||^K \Rightarrow ||\varphi||^{K'}$. K and K' are *bisimilar* iff exists a simulation ρ between K and K' such that ρ^{-1} is a simulation between K' and K. The set of all structures bisimilar to K is denoted by $\mathcal{B}(K)$. Bisimulation preserves CTL: $\forall \varphi \in$ CTL $\cdot \forall K' \in \mathcal{B}(K) \cdot ||\varphi||^K \Leftrightarrow ||\varphi||^{K'}$.

For a given a set of atomic propositions X, let K_{-X} denote the result of removing all atomic propositions in X from K, i.e., $AP_{-X} = AP \setminus X$. Let K and K' be Kripke structures such that $AP' = AP \cup X$. Then, K' is X*-bisimilar* to K iff K'_{-X} is bisimilar to K. The set of all X-bisimilar structures to K is denoted by $\mathcal{B}_X(K)$.

Definition 2. *[2] A relation* $\rho \subseteq S \times S'$ *is a* refinement *between 3-valued Kripke structures* K *and* K' *iff* $\rho(s, s')$ *implies*

1. $\forall p \in AP \cdot I(s,p) \preceq I'(s',p)$;
2. $\forall t \in S \cdot (R(s,t) \sqsupseteq \mathsf{true}) \Rightarrow \exists t' \in S' \cdot (R'(s',t') \sqsupseteq \mathsf{true}) \wedge \rho(t,t')$;
3. $\forall t' \in S' \cdot (R'(s',t') \sqsupseteq \mathsf{maybe}) \Rightarrow \exists t \in S \cdot (R(s,t) \sqsupseteq \mathsf{maybe}) \wedge \rho(t,t')$.

A state s is refined by s' ($s \preceq s'$) if there exists a refinement ρ containing (s, s'). A Kripke structure K is refined by K' ($K \preceq K'$) if there exists a refinement ρ relating their initial states: $\forall s \in S_0 \cdot \exists s' \in S_0' \cdot \rho(s,s')$ and $\forall s' \in S_0' \cdot \exists s \in S_0 \cdot \rho(s,s')$. Bisimulation and refinement coincide on classical structures, and refinement preserves 3-valued CTL:

Theorem 1. *[2] For 3-valued Kripke structures* K *and* K' *and a CTL formula* φ*,* $K \preceq K'$ *implies* $||\varphi||^K \preceq ||\varphi||^{K'}$.

Refinement can relate 3-valued and classical models as well. For a 3-valued Kripke structure K, let $\mathcal{C}(K)$ denote the set of *completions* [3] of K – the set of all classical Kripke structures that refine K. For any $K' \in \mathcal{C}(K)$, the structure K can be seen as less precise than K' in the sense that any CTL formula φ that evaluates to a definite value (either true or false) in K, evaluates to the same value in K', i.e., $(||\varphi||^K = \mathsf{true}) \Rightarrow (||\varphi||^{K'} = \mathsf{true})$ and $(||\varphi||^K = \mathsf{false}) \Rightarrow (||\varphi||^{K'} = \mathsf{false})$.

Thorough Semantics. Compositional semantics of CTL is inherently imprecise: if φ is maybe in K, it may or may not be true in every completion. To address this, Bruns and Godefroid [3] proposed an alternative semantics, calling it *thorough*. A formula φ is true in K under thorough semantics, written $||\varphi||_t^K = \mathsf{true}$, iff it is true in all completions of K; it is false in K if it is false in all completions; and maybe otherwise.

The additional precision comes at a cost of complexity. Model-checking φ under compositional semantics is linear in the size of the model and linear in the size of the formula, but model-checking φ under thorough semantics is EXPTIME-complete, with the best known algorithm quadratic in the size of the model and exponential in $|\varphi|$ [3].

Quantified CTL. Quantified CTL (QCTL) [19] is an extension of CTL with quantification over atomic propositions. Thus, QCTL formulas consist of all CTL formulas and formulas of the form $\forall x \cdot \varphi$ and $\exists x \cdot \varphi$. In this paper, we only use a fragment of QCTL in which all quantifiers precede all other operators. Thus, we consider formulas like $\forall x \cdot \exists y \cdot AG(x \Rightarrow AFy)$, but not like $AX(\exists x \cdot x \Rightarrow AFy)$, or $(\forall x \cdot EXx) \wedge (\exists y \cdot AXy)$

The syntax of QCTL does not restrict the domain of quantifiers. Thus, there are several different definitions of the semantics of QCTL with respect to a classical Kripke structure; we consider two of these in this paper: *structure* [19] and *amorphous* [10].

Structure Semantics. Under this semantics, each free variable x is interpreted as a boolean function over the statespace, i.e., $x \in [S \to \mathbf{2}]$. For example, $\forall x \cdot \varphi$ is true in K under structure semantics if replacing x by an arbitrary boolean function results in a formula that is true in K. Formally, the values of $\forall x \cdot \varphi$ and $\exists x \cdot \varphi$ over a Kripke structure K are defined as follows:

$$\begin{aligned} ||\varphi||_s^K &\triangleq ||\varphi||^K, \text{ if } \varphi \in \text{CTL} && \text{(structure semantics)} \\ ||\forall x \cdot \varphi||_s^K &\triangleq \forall y \in [S \to \mathbf{2}] \cdot ||\varphi[x \leftarrow y]||_s^K \\ ||\exists x \cdot \varphi||_s^K &\triangleq \exists y \in [S \to \mathbf{2}] \cdot ||\varphi[x \leftarrow y]||_s^K \end{aligned}$$

where $[S \to \mathbf{2}]$ denotes the set of all boolean functions over S.

Alternatively, structure semantics can be understood as follows. For Kripke structures K and K', we say that K' is an *X-variant* of K if there exists a set of atomic propositions X such that K and K'_{-X} are isomorphic. A formula $\forall x \cdot \varphi$ is satisfied by K under structure semantics iff φ holds in all $\{x\}$-variants of K. Note that if x is positive in φ, then $\forall x \cdot \varphi$ is equivalent to $\varphi[x \leftarrow \mathsf{false}]$, and if x is negative – to $\varphi[x \leftarrow \mathsf{true}]$.

Amorphous Semantics. Amorphous semantics of QCTL is defined as follows:

$$\begin{aligned} ||\varphi||_a^K &\triangleq ||\varphi||^K, \text{ if } \varphi \in \text{CTL} \qquad \text{(amorphous semantics)} \\ ||\forall x \cdot \varphi[x]||_a^K &\triangleq \forall K' \in \mathcal{B}_x(K) \cdot ||\varphi[x]||_a^{K'} \\ ||\exists x \cdot \varphi[x]||_a^K &\triangleq \exists K' \in \mathcal{B}_x(K) \cdot ||\varphi[x]||_a^{K'} \end{aligned}$$

That is, a formula $\forall x \cdot \varphi$ is satisfied by K under amorphous semantics iff φ is satisfied by every $\{x\}$-bisimulation of K.

For formulas without existential ($\exists$) quantifiers, amorphous semantics implies structure semantics; further, the implication is strict [10].

3 Expressiveness of 3-valued Models

In this section, we extend the results of Godefroid and Jagadeesan [13] on expressiveness of 3-valued models. In particular, we describe a transformation of 3-valued Kripke structures to Partial Kripke Structures (PKSs) – Kripke structures with boolean transition relation – and from there to Partial Classical Kripke Structures (PCKSs), where each atomic proposition is either always true or false, or is always maybe. This transformation enables us to use PCKSs as the theoretical model for developing our technical results. When compared to the original 3-valued Kripke structure, the transformation increases the number of atomic propositions. However, the transformation is used for theoretical purposes only – we never propose to apply this transformation during analysis. Furthermore, while increasing the number of atomic propositions, the transformation to PCKSs does not affect the number of bits required to encode the original Kripke structure.

From 3-valued models to PKSs. A 3-valued Kripke structure that has a boolean transition relation ($R : S \times S \to \mathbf{2}$) is called a *Partial Kripke Structure* (PKS) [2]. An example of a PKS is shown in Figure 5(a).

PKSs are as expressive as 3-valued Kripke structures [13]. The transformation T_1 from 3-valued to Partial Kripke structures is very similar to a transformation from Labeled Transition Systems to Kripke structures (e.g., see [21]). Intuitively, we treat transition values as actions, and the transformation "pushes" them into states.

Given a 3-valued Kripke structure K, we construct a PKS $T_1(K) = (AP \cup \{tval\}, S \times \{0,1\}, S_0 \times \{0,1\}, T_1(R), T_1(I))$, where $T_1(R)$ and $T_1(I)$ are as follows:

1. $T_1(R)(\langle s,i\rangle, \langle t,1\rangle) \Leftrightarrow (R(s,t) = \mathsf{true})$ and
   $T_1(R)(\langle s,i\rangle, \langle t,0\rangle) \Leftrightarrow (R(s,t) = \mathsf{maybe})$,
2. for every $p \in AP$, $T_1(I)(\langle s,i\rangle, p) = I(s,p)$, and
3. the value of $tval$ is determined by the second component of the state:
   $T_1(I)(\langle s,i\rangle, tval)$ is true if $i = 1$, and maybe otherwise.

$$
\begin{array}{ll}
T_1(p) = p & T_1(\neg\varphi) = \neg T_1(\varphi) \\
T_1(\varphi \wedge \psi) = T_1(\varphi) \wedge T_1(\psi) & T_1(\varphi \vee \psi) = T_1(\varphi) \vee T_1(\psi) \\
T_1(EX\varphi) = EX(tval \wedge T_1(\varphi)) & T_1(EG\varphi) = \varphi \wedge EXEG(tval \wedge T_1(\varphi))
\end{array}
$$
$$T_1(E[\varphi U \psi]) = T_1(\psi) \vee T_1(\varphi) \wedge EXE[tval \wedge T_1(\varphi)\ U\ tval \wedge T_1(\psi)]$$

Fig. 4. Transformation of a temporal logic formula.

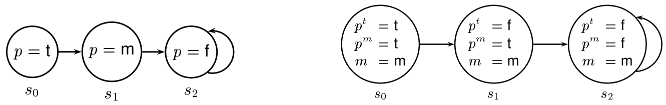


Fig. 5. (a) A PKS K. (b) A PCKS K'.

Intuitively, $tval$ represents the value of the transition relation. For example, since the value of $tval$ in a state $\langle t, 1\rangle$ is true, a transition between $\langle s, i\rangle$ and $\langle t, 1\rangle$ indicates that the transition between s and t in K is true.

The transformation T_1 is also extended to CTL formulas as shown in Figure 4. Intuitively, T_1 replaces every occurrence of EXp with $EX(tval \wedge p)$ in the fixpoint representation of the semantics of CTL (see Figure 3).

Theorem 2. *[13] Partial Kripke Structures are as expressive as 3-valued Kripke structures. For any 3-valued Kripke structure K and a formula φ, $||\varphi||^K = ||T_1(\varphi)||^{T_1(K)}$.*

From PKSs to PCKSs. A PKS in which every atomic proposition is either boolean (i.e., true or false in every state) or maybe (i.e. maybe in every state) is called a *Partial Classical Kripke Structure* (PCKS), an example of a PCKS is shown in Figure 5(b). Intuitively, a PCKS K is a classical Kripke structure extended with additional atomic propositions such that nothing except their name is known about them. We show that, for compositional semantics, PCKSs containing a *single* maybe atomic proposition are as expressive as PKSs.

A value of a propositional formula in a 3-valued Kripke structure is given by a 3-valued function $S \to \mathbf{3}$ over the statespace. Consider a PKS K shown in Figure 5(a). The value of p in K is given by a function that maps s_0 to true, s_1 to maybe, and s_3 to false. Next, consider the PCKS K' shown in Figure 5(b): it is the same structure, but with different atomic propositions. All atomic propositions of K' are boolean, except for m which is maybe in every state. Note that K' has two boolean atomic propositions p^t and p^m such that p^t is true in a state iff p is true in the same state of K, and p^m is true iff p is not false. The formula $p^t \vee (p^m \wedge m)$ in K' is semantically equivalent to p in K: for any state, both are true in s_1, maybe in s_2, and false in s_3. Thus, any propositional formula in K can be reduced to a semantically equivalent one in K'. Furthermore, temporal operators of CTL can be seen as predicate transformers operating on the semantic meaning of their arguments. Thus, the value of EXp in K is equivalent to the value of $EX(p^t \vee (p^m \wedge m))$ in K'.

Formally, we define a transformation T_2 from a PKS K to a PCKS $T_2(K) = (T_2(AP), S, S_0, R, T_2(I))$ as follows: (a) for each atomic proposition p of K, $T_2(AP)$ contains a pair of boolean atomic propositions p^t and p^m, (b) $T_2(I)(s, p^t)$ is true iff

$I(s,p)$ is true, and $T_2(I)(s,p^m)$ is true iff $I(s,p)$ is *not* false, and (c) $T_2(AP)$ contains an atomic proposition m whose value is maybe in every state of $T_2(K)$.

For an atomic proposition p, $T_2(p)$ is defined as $p^t \vee (p^m \wedge m)$, and for a CTL formula φ, $T_2(\varphi)$ is obtained by replacing each atomic proposition p of φ with $T_2(p)$. For example, $T_2(AG(p \Rightarrow EFq)) = AG(T_2(p) \Rightarrow EFT_2(q))$.

Theorem 3. *Let K be a PKS, and φ be a CTL formula. Then,* $||\varphi||^K = ||T_2(\varphi)||^{T_2(K)}$.

Combining this result with Theorem 2, we obtain that PCKSs are as expressive (for compositional semantics) as 3-valued Kripke structures.

The transformation T_2 does not work in the case of thorough semantics: the value of φ in K is not necessarily equivalent to the value of $T_2(\varphi)$ in $T_2(K)$. For example, under thorough semantics, the value of $p \vee \neg q$ is maybe in a state where both p and q are maybe. However, since $p =$ maybe implies $p^t =$ false and $p^m =$ true, the transformed formula $T_2(p \vee \neg q) = (p^t \vee (p^m \wedge m)) \vee \neg(q^t \vee (q^m \wedge m))$ is logically equivalent to $m \vee \neg m$, which, in turn, is equivalent to true under thorough semantics. The problem is that in each state of $T_2(K)$, the atomic proposition m controls how *all* of the atomic propositions in this state are refined (i.e., either they are all set to true, or they are all set to false). This is easily avoided by introducing a different atomic proposition for each atomic proposition of K.

We define another transformation T_3 from a PKS K to a PCKS $T_3(K)$ as follows: (a) we first apply the transformation T_2, i.e., $T_3(K) = T_2(K)$, and (b) for each $p \in AP$ we add a new atomic proposition m_p to $T_3(AP)$, setting it to maybe in every state. For an atomic proposition p, $T_3(p)$ is defined as $p^t \vee (p^m \wedge m_p)$, and for a CTL formula φ, $T_3(\varphi)$ is obtained by replacing each atomic proposition p of φ with $T_3(p)$.

Theorem 4. *Let K be a PKS, and φ be a CTL formula. Then,* $||\varphi||_t^K = ||T_3(\varphi)||_t^{T_3(K)}$.

Combining this result with Theorems 2 and 3, we obtain that PCKSs are as expressive as 3-valued Kripke structures, for compositional and thorough semantics.

The distinction between transformations T_2 and T_3 highlights the key difference between compositional and thorough semantics. The former can be seen as a conservative approximation of laws of excluded middle and non-contradiction, i.e., if p is unknown, then so is $\neg p$, and thus $||p \vee \neg p|| =$ maybe $\vee$ maybe $=$ maybe. On the other hand, thorough semantics can be seen as applying these laws symbolically. Thus, even if the value of p is unknown, $||p \vee \neg p||_t$ is still true.

4 Quantified Temporal Logic and 3-valued Model-Checking

In this section, we use the equivalence between 3-valued Kripke structures and PCKSs established in Section 3 to relate 3-valued model-checking and model-checking for QCTL.

The definition of 3-valued refinement, when restricted to PCKSs, is virtually identical to the definition of X-bisimulation. If K is a PCKS and X is the set of *all* of its maybe atomic propositions, then K' is a completion of K iff K'_{-X} is bisimilar to K_{-X}, i.e., K' is X-bisimilar to K_{-X}. Thus, deciding whether a formula φ is either true or false in a PCKS reduces to amorphous model-checking of a universally quantified formula, as stated in the theorem below.

Theorem 5. *Let K be a PCKS, $X \subseteq AP$ be the set of all of its* maybe *atomic propositions, and φ be an arbitrary CTL formula. Then, the value of φ in K under thorough semantics is:* $(||\varphi||_t^K = \mathsf{true}) \Leftrightarrow ||\forall X \cdot \varphi||_a^{K-x}$ *and* $(||\varphi||_t^K = \mathsf{false}) \Leftrightarrow ||\forall X \cdot \neg\varphi||_a^{K-x}$.

Similarly, compositional semantics is related to structure semantics for QCTL; however, the connection is somewhat more subtle. Let K be a PCKS, m be the *only* maybe atomic proposition of K, and φ be a CTL formula containing m. Furthermore, assume that all occurrences of m are positive. Then, $||\varphi||^K$ is true iff $||\varphi[m \leftarrow \mathsf{false}]||^{K-m}$ is true [16]. Next, consider the formula $\forall m \cdot \varphi$: since m is positive in φ, $||\forall m \cdot \varphi||_s^{K-m}$ is true iff $||\varphi[m \leftarrow \mathsf{false}]||^{K-m}$ is true [17]. Thus, in this case, deciding whether φ is true under compositional semantics reduces to checking $\forall m \cdot \varphi$ under structure semantics. Moreover, the result easily extends to the case where m occurs negatively.

The above does not hold when m is not of pure polarity in φ. For example, the value of $||m \vee \neg m||^K$ is maybe, but $||\forall m \cdot (m \vee \neg m)||_s^K$ is true. The problem is that compositional semantics treats positive and negative occurrences of the same atomic proposition independently. Thus, we can obtain the desired result by quantifying positive and negative occurrences of m separately. That is, we let m^+ and m^- denote positive and negative occurrences of m in $\varphi[m]$, respectively; then, $||\varphi[m]||^K$ is true iff $||\forall x, y \cdot \varphi[m^+ \leftarrow x, m^- \leftarrow y]||_s^K$ is true, and similarly $||\varphi[m]||^K$ is false iff $||\forall x, y \cdot \neg\varphi[m^+ \leftarrow x, m^- \leftarrow y]||_s^K$ is true. The following theorem formalizes this result for an arbitrary number of maybe atomic propositions.

Theorem 6. *Let K be a PCKS, and let $M = \{m_1, \ldots, m_n\}$ be the set of all* maybe *atomic propositions of K. For a CTL formula φ, let m_i^+ and m_i^- denote the positive and negative occurrences of m_i, respectively. Then,*

$$\begin{array}{l}(||\varphi||^K = \mathsf{true}) \;\Leftrightarrow ||\forall x_1, \ldots, x_n \cdot \forall y_1, \ldots, y_n \cdot \varphi'||_s^K \textit{ and} \\ (||\varphi||^K = \mathsf{false}) \Leftrightarrow ||\forall x_1, \ldots, x_n \cdot \forall y_1, \ldots, y_n \cdot \neg\varphi'||_s^K\end{array}$$

where $\varphi' = \varphi[m_1^+ \leftarrow x_1, \ldots, m_n^+ \leftarrow x_n, m_1^- \leftarrow y_1, \ldots, m_n^- \leftarrow y_n]$.

A corollary of Theorem 6 is that if every maybe atomic proposition of K occurs with pure polarity in φ, then both thorough and compositional semantics reduce to deciding the same universally quantified formula, under amorphous and structure semantics, respectively. Furthermore, for universally quantified formulas, amorphous semantics imply structure ($||\forall X \cdot \varphi||_a \Rightarrow ||\forall X \cdot \varphi||_s$). Note that in general, for 3-valued semantics the implication is reversed, i.e., compositional semantics implies thorough $((||\varphi||^K = \mathsf{true}) \Rightarrow (||\varphi||_t^K = \mathsf{true})$. So, when every maybe atomic proposition is pure in φ, thorough and compositional semantics for φ coincide:

Theorem 7. *Let K be a PCKS and φ be a CTL formula such that all occurrences of* maybe *atomic propositions of K are of pure polarity in φ. Then, $||\varphi||_t^K = ||\varphi||^K$.*

Since every atomic proposition is either boolean or maybe in PCKSs, deciding whether all occurrences of maybe propositions in a formula φ are of pure polarity is trivial for these models. However, to determine this for arbitrary 3-valued Kripke structures, we first have to reduce them to PCKSs, which is not an option in practice since model-checking typically occurs on-the-fly during the construction of the model. In the next section, we use properties of particular abstractions to determine polarity of maybe propositions of φ and thus to decide when a thorough check is necessary.

5 Thorough Semantics and Abstraction

In this section, we exhibit practical cases where a thorough check does not give additional precision and thus can be eliminated, and cases where a thorough check can be performed efficiently.

5.1 Abstraction and 3-Valued Model Checking

Abstraction is a mapping between a concrete system and a smaller, abstracted, system. Here, we consider abstractions that map sets of concrete states into a single abstract state. Let K be a Kripke structure with statespace S and transition relation R. An *abstract domain* is a pair (S_α, γ), where S_α is a set of abstract states, and $\gamma : S_\alpha \to 2^S$ is a total *concretization function* that associates each abstract state with its interpretation as a set of concrete states.

Like Godefroid et al. [11], we use 3-valued Kripke structures to represent abstract models over an abstract domain (S_α, γ). A 3-valued Kripke structure K_α with a statespace S_α is an abstraction of a Kripke structure K if its transition relation R_α satisfies the following conditions:

$$
\begin{aligned}
(\mathbb{R}_\alpha(\hat{s}, \hat{t}) \sqsupseteq \mathsf{true}) &\Rightarrow \forall s \in \gamma(\hat{s}) \cdot \exists t \in \gamma(\hat{t}) \cdot R(s,t) \\
(\mathbb{R}_\alpha(\hat{s}, \hat{t}) \sqsupseteq \mathsf{maybe}) &\Leftarrow \exists s \in \gamma(\hat{s}) \cdot \exists t \in \gamma(\hat{t}) \cdot R(s,t)
\end{aligned}
$$

Note that these conditions do not guarantee the precision of the abstract model. In particular, a 3-valued Kripke structure over S_α with a maybe transition between every pair of states satisfies the above conditions, and is a trivial abstraction of every classical Kripke structure over S.

Each atomic proposition of K_α corresponds to a predicate over the statespace of K. In an abstract state $\hat{s}$, an atomic proposition $\hat{p}$ is true iff the corresponding predicate p is true in every state of $\gamma(\hat{s})$, false if p is false in $\gamma(\hat{s})$, and maybe otherwise. Note that any predicate over the concrete statespace can be replaced by an atomic proposition. Thus, without loss of generality, we assume that every atomic proposition of the abstract system corresponds to an atomic proposition of the concrete.

As a 3-valued Kripke structure, an abstraction K_α of K is refined by K, i.e., $K_\alpha \preceq K$, which guarantees that K_α preserves arbitrary CTL formulas. Moreover, an arbitrary 3-valued Kripke structure is an abstraction of any model that refines it, where the concretization γ is induced by the refinement [11].

Predicate (or boolean) abstraction [14, 1, 11] is a popular technique for building abstractions, and has been successfully applied in practice [14, 5]. Given a concrete system K and a set of n predicates $P = \{p_1, \ldots, p_n\}$, the abstract statespace of predicate abstraction consists of (at most) 2^n states, where each state assigns a boolean value to each of the predicates. The concretization γ is defined as follows:

$$\gamma(\hat{s}) = \{s \mid \forall p \in P \cdot ||p||(\hat{s}) = ||p||(s)\}$$

That is, an abstract state $\hat{s}$ corresponds to the set of all concrete states that agree with $\hat{s}$ on the values of all of the predicates in P. Thus, if K_α is a result of predicate abstraction, then its transition relation is 3-valued, but atomic propositions are boolean.

Cartesian abstraction [1, 11] is an extension of predicate abstraction, where the statespace consists of 3^n states, and each state assigns one of true, false, or maybe to each of the predicates. The concretization γ is defined as follows:

$$\gamma(\hat{s}) = \{s \mid \forall p \in P \cdot ||p||(\hat{s}) \preceq ||p||(s)\}$$

That is, an abstract state $\hat{s}$ corresponds to the set of all concrete states that agree with $\hat{s}$ on the values of all of the predicates in P that have a definite value (i.e. true or false) in $\hat{s}$. Thus, if K_α is a result of a Cartesian abstraction, then both its atomic propositions and the transition relation are 3-valued.

Model-checking a property φ in the abstract system K_α is done with respect to compositional semantics. Thus, a maybe result from the model-checker does not necessarily indicate that the abstraction is at fault and must be refined. In these cases, it seems natural [12] that an additional check of φ under thorough semantics will yield more precise results. In what follows, we show that in many practical applications, thorough semantics does not offer an advantage over compositional.

5.2 Thorough Semantics and Predicate Abstraction

Let K_α be an abstract system constructed by predicate abstraction, and $K'_\alpha = T_1(K_\alpha)$ be a PKS corresponding to it. Note that all of the atomic propositions of K'_α are boolean, except for $tval$, which was added as part of T_1.

Assume that we want to check a CTL formula φ in K_α. By Theorem 2, there exists a CTL formula $\varphi' = T_1(\varphi)$ such that $||\varphi||^{K_\alpha} = ||\varphi'||^{K'_\alpha}$. Although φ does not mention $tval$ explicitly, each temporal operator of φ results in at least one occurrence of $tval$ in φ'. The polarity of these occurrences is positive for existential operators and negative for the universal ones. For example, EXp is transformed by T_1 into $EX(tval \wedge p)$, while AXp is transformed into $T_1(AXp) = T_1(\neg EX\neg p) = AX(tval \Rightarrow p)$.

Thus, if all temporal operators of φ are either universal or existential, i.e., $\varphi \in$ ECTL or $\varphi \in$ ACTL, then φ' contains at most one non-boolean atomic proposition $tval$, and $tval$ is pure in φ'. Combining this with Theorem 7, we establish that in this case thorough and compositional semantics for φ in K_α coincide:

Theorem 8. *Let K_α be a 3-valued Kripke structure constructed by predicate abstraction. Then, $\forall \varphi \in$ ECTL $\cup$ ACTL $\cdot$ $||\varphi||^{K_\alpha} = ||\varphi||_t^{K_\alpha}$.*

In particular, this theorem implies that for predicate abstraction and for universal properties, the original abstraction-refinement framework of Clarke et al. [5] is as precise as the extension proposed by Godefroid and Jagadeesan [12].

In the case of Cartesian abstraction, K_α may contain 3-valued atomic propositions, and Theorem 8 is no longer applicable. One way to ensure that thorough and compositional semantics coincide in this case, is to require that all atomic propositions, not just $tval$, be of pure polarity. This gives rise to the following theorem:

Theorem 9. *Let K_α be a 3-valued Kripke structure. Then, for any ACTL or ECTL formula φ in which every atomic proposition occurs with pure polarity, compositional and thorough semantics are equivalent.*

For example, according to the above theorem, compositional and thorough semantics of $AG(\neg p \wedge q)$ are equivalent, since each atomic proposition occurs once, and polarity of p is negative, and polarity of q is positive. Of course, many interesting properties do contain atomic propositions of mixed polarity. For example, a property "in every state, only one of p and q holds" is expressed in CTL as $AG((\neg p \wedge q) \vee (p \wedge \neg q))$, and both of its atomic propositions are of mixed polarity. In this case, thorough semantics can offer additional precision. On the other hand, consider checking the property $AG(\neg q \wedge AF(p \wedge q))$ on the model in Figure 2(b). In this property, q occurs with mixed polarity, but it does not have value maybe in any reachable state of the model. For this and other properties where the proposition of mixed polarity does not have value maybe in the model, compositional semantics coincides with thorough, and the additional check is not required.

5.3 Thorough Model Checking for ACTL

In this section, we show that in the case of ACTL formulas, which are sufficient for expressing arbitrary safety properties, deciding whether a formula is true under thorough semantics can be done efficiently. Furthermore, in this case, the compositional check used in the abstraction-refinement framework of Clarke et al. [5] can be completely replaced by an efficient algorithm for implementing the thorough one.

We start by showing that for a classical Kripke structure and an ACTL formula φ, model-checking $\forall x \cdot \varphi[x]$ under amorphous semantics is reducible to model-checking $\varphi[x]$ (I). Using Theorem 5, we extend this result to an efficient algorithm for deciding whether an ACTL formula is true under thorough semantics on a PKS (II), and, finally, doing the same on an arbitrary 3-valued Kripke structure (III).

(I). Let K be a classical Kripke structure, x be an atomic proposition that does not appear in K, and φ be an ACTL formula containing x. Recall that $||\forall x \cdot \varphi||_a^K$ is true iff φ is true in every K' that is $\{x\}$-bisimilar to K. Let $T_4(K) = (T_4(AP), T_4(S), T_4(S_0), T_4(R), T_4(I))$ be a Kripke structure obtained from K by adding a new atomic proposition x that changes non-deterministically. The transformation T_4 is defined as follows:

$$\begin{aligned} T_4(AP) &= AP \cup \{x\} \\ T_4(S) &= S \times \{0,1\} \\ T_4(S_0) &= S_0 \times \{0,1\} \\ T_4(R)(\langle s,i\rangle,\langle t,j\rangle) &\Leftrightarrow R(s,t) \\ T_4(I)(\langle s,i\rangle,p) &= I(s,p) \\ T_4(I)(\langle s,i\rangle,x) &= \mathsf{true} \text{ if } i=1 \text{ and} \\ & \quad\; \mathsf{false} \text{ otherwise} \end{aligned}$$

Note that the value of each atomic proposition $p \in AP$ is determined by the first component of the state, and the value of x depends on the second component.

Clearly, $T_4(K)$ is $\{x\}$-bisimilar to K. Moreover, any Kripke structure that is $\{x\}$-bisimilar to K is *simulated* by $T_4(K)$ [17]. Since simulation preserves ACTL, $||\forall x \cdot \varphi||_a^K$ is equivalent to $||\varphi||^{T_4(K)}$. The result easily extends to an arbitrary number of universally-quantified atomic propositions of φ. Note that if x is of pure polarity in φ, the transformation T_4 is unnecessary, since $||\forall x \cdot \varphi[x]||_a^K$ is equivalent to either $||\varphi[x \leftarrow \mathsf{true}]||^K$ or $||\varphi[x \leftarrow \mathsf{false}]||^K$, depending on the polarity of x.

(II). Combining (I) with Theorem 5, we conclude that deciding whether an ACTL formula φ is true in a PKS K under thorough semantics is reducible to classical model-checking. In particular, for an ACTL formula φ, $||\varphi||_t^K = \mathsf{true}$ iff $||\varphi||^{K'} = \mathsf{true}$, where K' is a classical Kripke structure obtained from K via a process very similar to T_4, treating maybe atomic propositions non-deterministically. However, rather than splitting all states, we only split those where an atomic proposition has a value maybe. That is, if p is an atomic proposition and s is a state such that the value of p in s is maybe, then s is replaced by two states s' and s'' such that

(a) s' and s'' have the same successors as s,
(b) for every atomic proposition q different from p, s' and s'' assign the same interpretation as s ($I(s,q) = I(s',q) = I(s'',q)$),
(c) the value of p is true in s' and false in s'', and
(d) every transition from a state t to s is replaced by a pair of transitions from t to s' and s''.

This process is repeated until there are no more reachable states that assign maybe to an atomic proposition. Since each atomic proposition that is treated non-deterministically doubles the statespace, the statespace of K' is in the worst case exponential in the number of atomic propositions of K.

(III). From amorphous semantics, we know that $\forall x \cdot \varphi$ is equivalent to $\varphi[x \leftarrow \mathsf{false}]$ if x is positive in φ, and to $\varphi[x \leftarrow \mathsf{true}]$ if x is negative. Therefore, our translation can treat atomic propositions that are of pure polarity in φ as either true or false, depending on the polarity, whereas others must be treated non-deterministically. Thus, for a 3-valued Kripke structure K and an ACTL formula φ, deciding whether $||\varphi||_t^K = \mathsf{true}$ is reducible to model-checking φ in K', obtained from K as follows:

(a) for every positive atomic proposition of φ, change its maybe occurrences in K to false,
(b) change maybe occurrences of negative ones to true,
(c) treat mixed ones as non-deterministic, and
(d) change all maybe transitions to true.

Note that transitions can be embedded into states using an atomic proposition $tval$ (see Section 3), which has negative polarity for ACTL. In the worst case, the size of K' is exponential only in the number of mixed atomic propositions of φ, which gives our algorithm the following complexity:

Theorem 10. *Let K be a 3-valued Kripke structure, and φ be an ACTL formula. Then, the complexity of deciding whether φ is* true *in K under thorough semantics is $O(2^n \times |K| \times |\varphi|)$, where n is the number of atomic propositions of mixed polarity in φ.*

Since we reduced the thorough check to classical model-checking, our algorithm either produces a definite result or generates a counterexample. Thus, it can completely replace step 3 in the abstraction refinement framework of Clarke et al. [5], shown in Figure 1(a). The resulting framework is as precise as the *classical thorough* framework (see Section 1 for definition), and requires the same number of iterations. Yet it is only marginally more expensive than the original framework. Moreover, in the case where all atomic propositions of φ are pure, the modified framework is the same as the original: same results, same running time. Finally, the algorithm can be applied on-the-fly, i.e., during the construction of the abstract model.

6 Discussion and Related Work

Dams et at. [9] developed a general framework for constructing abstractions based on the Abstract Interpretation [8] methodology. These abstractions are sound for full CTL (and richer logics such as CTL^* and μ-calculus). Instead of 3-valued Kripke structures, their modeling formalism is Mixed Transition Systems (MTSs) – transition systems containing two kinds of transitions, where existential path quantifiers are interpreted over one kind and universal over the other. 3-valued Kripke structure can be seen as MTSs where truth of existential path quantifiers depends only on true transitions, while the truth of universal quantifiers depends on both true and maybe transitions [16].

The work of Dams et al. [9], as well as most other research on combining abstraction and model-checking (e.g., see [5, 22, 14]), handles explicit occurrences of negation in a formula by restricting negation to the level of atomic propositions and treating each literal of the concrete model as an atomic proposition of the abstract. For example, literals p and $\neg p$ are represented by two *distinct* atomic propositions, say, a and b. This looses information but ensures that all of the atomic propositions of a formula checked on an abstract model are pure, and thus a thorough check does not provide an additional advantage.

Thorough semantics was introduced by Bruns and Godefroid [3] via *generalized model-checking*, which is the problem of deciding whether there exists a completion of a 3-valued Kripke structure in which a given formula holds. This can be seen as a generalization of both satisfiability and model-checking: φ is true in the coarsest abstraction iff φ is satisfiable, and true in a classical Kripke structure K iff K is a model for φ. In this paper, we show that generalized model-checking can be also seen as an extension of amorphous semantics for existentially quantified temporal formulas from PCKSs to arbitrary 3-valued Kripke structures. In a sense, it combines amorphous quantification with the reduction to PCKSs.

The expressive power of various 3-valued models have been studied by Godefroid and Jagadeesan [13]. Our work completes the picture by showing that allowing for maybe atomic propositions is as expressive as allowing unrestricted occurrences of the value maybe in a model. The question whether or not 3-valued Kripke structures with boolean atomic propositions and a 3-valued transition relation are as expressive remains open. However, our results suggest that even if such a reduction exists, it is not trivial. In particular, this reduction would allow us to transform model-checking of ACTL under thorough semantics, which is EXPTIME-complete, into model-checking under compositional semantics, which is linear in the size of the model and the formula.

7 Conclusion

In this paper, we study the difference between compositional and thorough semantics for 3-valued model-checking. We show that the relationship between the two becomes more clear by casting 3-valued model-checking as model-checking for quantified temporal logic.

Our main motivation is a seemingly apparent advantage of thorough semantics over compositional in the abstraction refinement framework. However, we show that in many practically interesting cases, i.e., when properties are universal, thorough semantics is either no more precise than compositional, or can be efficiently combined with classical model-checking approaches. Although we used CTL as our temporal logic, our results depend only on its invariance to bisimulation, and thus naturally extend to other universal logics, such as LTL.

References

1. T. Ball, A. Podelski, and S. Rajamani. “Boolean and Cartesian Abstraction for Model Checking C Programs”. In *TACAS'01*, vol. 2031 of *LNCS*, pp. 268–283, 2001.
2. G. Bruns and P. Godefroid. “Model Checking Partial State Spaces with 3-Valued Temporal Logics”. In *CAV'99*, vol. 1633 of *LNCS*, pp. 274–287, 1999.
3. G. Bruns and P. Godefroid. “Generalized Model Checking: Reasoning About Partial State Spaces”. In *CONCUR'00*, vol. 1877 of *LNCS*, pp. 168–182, 2000.
4. M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. “Multi-Valued Symbolic Model-Checking”. *ACM Trans. on Soft. Eng. and Methodology*, 12(4):1–38, 2003.
5. E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. “Counterexample-Guided Abstraction Refinement for Symbolic Model Checking”. *Journal of the ACM*, 50(5):752–794, 2003.
6. E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
7. E.M. Clarke, E.A. Emerson, and A.P. Sistla. “Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications”. *ACM Trans. on Prog. Lang. and Systems*, 8(2):244–263, 1986.
8. P. Cousot and R. Cousot. “Abstract Interpretation: A Unified Lattice Model For Static Analysis of Programs by Construction or Approximation of Fixpoints”. In *POPL'77*, pp. 238–252, Los Angeles, California, 1977.
9. D. Dams, R. Gerth, and O. Grumberg. “Abstract Interpretation of Reactive Systems”. *ACM Trans. on Prog. Lang. and Systems*, 2(19):253–291, 1997.
10. T. French. “Decidability of Quantifed Propositional Branching Time Logics”. In *AI'01*, vol. 2256 of *LNCS*, pp. 165–176, 2001.
11. P. Godefroid, M. Huth, and R. Jagadeesan. “Abstraction-Based Model Checking Using Modal Transition Systems”. In *CONCUR'01*, vol. 2154 of *LNCS*, pp. 426–440, 2001.
12. P. Godefroid and R. Jagadeesan. “Automatic Abstraction Using Generalized Model-Checking”. In *CAV'02*, vol. 2404 of *LNCS*, pp. 137–150, 2002.
13. P. Godefroid and R. Jagadeesan. “On the Expressiveness of 3-Valued Models”. In *VMCAI'03*, vol. 2575 of *LNCS*, pp. 206–222, 2003.
14. S. Graf and H. Saïdi. “Construction of Abstract State Graphs with PVS”. In *CAV'97*, vol. 1254 of *LNCS*, 1997.
15. A. Gurfinkel and M. Chechik. “Generating Counterexamples for Multi-Valued Model-Checking”. In *FME'03*, vol. 2805 of *LNCS*, 2003.
16. A. Gurfinkel and M. Chechik. “Multi-Valued Model-Checking via Classical Model-Checking”. In *CONCUR'03*, vol. 2761 of *LNCS*, 2003.
17. A. Gurfinkel and M. Chechik. “Extending Extended Vacuity”. In *FMCAD'04*, vol. 3312 of *LNCS*, pp. 306–321, 2004.
18. S. C. Kleene. *Introduction to Metamathematics*. New York: Van Nostrand, 1952.
19. O. Kupferman. “Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions”. *J. of Logic and Computation*, 7:1–14, 1997.
20. R. Milner. ”An Algebraic Definition of Simulation between Programs”. In *AI'71*, pp. 481–489, 1971.
21. M. Müller-Olm, D. Schmidt, and B. Steffen. “Model-Checking: A Tutorial Introduction”. In *SAS'99*, vol. 1694 of *LNCS*, pp. 330–354, 1999.
22. S. Shoham and O. Grumberg. “A Game-Based Framework for CTL Counter-Examples and 3-Valued Abstraction-Refinement”. In *CAV'03*, vol. 2725 of *LNCS*, pp. 275–287, 2003.