Model checking modal transition systems using Kripke structures

Michael Huth

Department of Computing, Imperial College of Science, Technology and Medicine,
mrh@doc.ic.ac.uk

Abstract. We reduce the modal mu-calculus model-checking problem for Kripke modal transition systems to the modal mu-calculus model-checking problem for Kripke structures. This reduction is sound, preserves the alternation-depth fragments of the modal mu-calculus, is linear in the size of formulas and models, and extends the reach of modal mu-calculus model checkers to sound abstraction for the full logic. These results specialize to CTL* model-checking and CTL model checking.

1 Introduction

Model-based property verification of software inescapably has to mitigate computational complexities whose roots are the concurrent interaction of communicating programs and the size and structure of data types. Abstraction is widely recognized as a key technology in containing these complexities (e.g. [2, 10, 12, 15, 18, 22, 31, 32]). Since todays software depends on a high degree of communication and reactiveness, property verification can only succeed if reasonable assumptions are being made about thread scheduling, the access policies to resources, progress conditions on communication, etc. Filter-based refinement [19] and fairness conditions [21] are well established and widely practiced approaches of formalizing and enforcing such additional assumptions about the interaction of software systems. The model checker SMV [26], for example, supports simple fairness constraints [8, 9] that reduce state-space exploration to those paths on which a finite number of CTL formulas hold infinitely often. The modal mu-calculus of alternation depth 2 serves as a target specification language that can express (branching-time) filter-based refinement, CTL, CTL*, CTL with simple fairness constraints, and many other properties that need to be expressible in the verification of reactive software, such as "event p occurs at every other state" [20]. Although that fragment ($k = 2$) delineates, for most practitioners, the realm of property-verification applications, we present our technical work on the abstraction-based verification of concurrent, reactive software for arbitrary alternation-depth fragments ($k \geq 0$) of the modal mu-calculus. In [16], it has been recognized that the sound use of fairness assumptions for abstraction-based reasoning has to be exercised with some care and a sound three-valued solution for fair CTL* has been given. Although universal and existential properties can each be soundly abstracted with a corresponding notion of simulation

[29, 33], more complex notions of refinement are required for the sound abstraction of their combinations [14, 31, 13] — examples being the symbolic encoding of CTL model checking with simple fairness constraints [8], and the use of logical implication for filter-based refinement in a branching-time logic.

In this paper, we re-examine this delicate but important combination of abstraction techniques and property verification for the modal mu-calculus and its alternation depth fragments in general. We use Kripke modal transition systems [23] (Kripke MTSs) as our designated models for abstraction-based model checking [22] which — being three-valued versions of doubly-labeled transition systems [17] — are expressive models for under-specified, or under-determined systems. The intent of our work is predominantly pragmatic in nature in that we mean to reduce the model-checking problem for Kripke MTSs to existing ones (Kripke structures), allowing the instrumented re-use of tools. Kripke MTSs are designed to guarantee soundness for abstraction-based model checking of arbitrary formulas of the modal mu-calculus [23].[1] This class of models encompasses important classes of qualitative models that have three-valued specifications — be they on transitions [27], state propositions [6] or expressed through divergence [30]. Consequently, a model checking reduction for Kripke MTSs applies to these models as well; see Figure 1.[2] The Kripke structures computed in our model-checking reduction can be described and checked (for CTL) in tools such as SMV [26] and extensions of Spin, e.g. tools that implement non-emptiness checking for hesitant alternating automata [36].

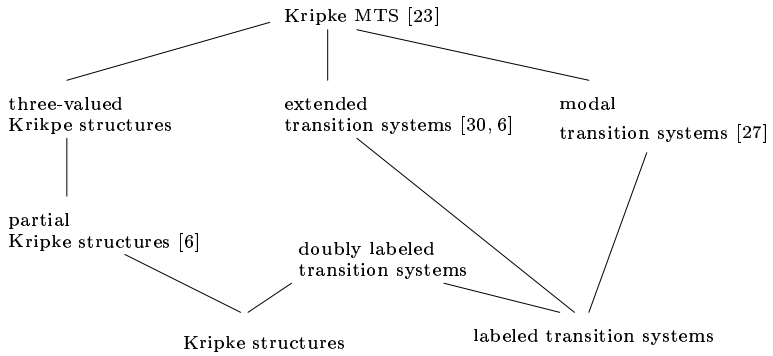

Fig. 1. A Hasse diagram of classes of two-valued (doubly labeled transition systems, Kripke structures, and labeled transition systems) and three-valued (all other classes) models, where the order represents class inclusion up to isomorphism.

[1] Similar guarantees have already been established for quantified logics with negation, e.g [6, 13, 31].

[2] Three valued Kripke structures are Kripke MTS with a sole action type.

Main results. The principal contributions of this paper are that we reduce the model-checking problem of the modal mu-calculus for Kripke MTSs to a model-checking problem of the modal mu-calculus for Kripke structures with an extended/collapsed signature. This reduction is linear in the size of models and formulas. For formulas, it leaves their entire recursion structure, and therefore their alternation depth, invariant. In particular, a model check of a Kripke MTS turns out to be no more complex than a model check of the resulting Kripke structure. Although our results apply to all fragments of the modal mu-calculus, the fragments of alternation depth 1 and 2 have practical importance. For example, for alternation depth 1, we get a reduction of model checking CTL over Kripke MTSs to model checking CTL over Kripke structures, at no additional cost.

Related work. In [7], Bruns & Godefroid pioneered such a programme for partial Kripke structures [6] (see Figure 1), which are three-valued versions of Kripke structures in that atomic propositions at states are either false, true, or undetermined. They transform such a structure into two Kripke structures and a model mu-calculus formula into one in positive normal form. Their model-checking reduction is sound and complete and does not increase the size of the models, nor the cost of the model check. We merely generalize such a result to the class of Kripke MTSs, which subsumes the class of partial Kripke structures and other three-valued classes of models (see Figure 1). Although it is possible that there is a direct translation from Kripke MTSs into partial Kripke structures, we do not know of any one in the literature. Even if such a translation exists, it is of interest to study the explicit nature of alternative translations, notably with respect to their capacity of preserving the "modalities" of paths (must-paths and may-paths [34]) — which would allow for the algorithmic separation of fairness constraints — and their ability of providing useful debugging information. Our reduction proceeds in two stages, one of which is a straightforward adaptation of a reduction of model-checking modal transition systems to model-checking labeled transition systems [22].

Outline of paper. In Section 2, we define doubly labeled transition systems (DLTSs) and Kripke MTSs, and two, mutually recursive, property semantics for Kripke MTSs — one for assertion checks and one for consistency checks. We mention that assertion checks on abstract Kripke MTSs are sound for all formulas of the underlying logic and that both semantics preserve the usual DeMorgan laws. Section 3 presents, for sake of illustration, the usual encoding of CTL with simple fairness constraints in the modal mu-calculus of alternation depth 2, and notes that its conversion into positive normal form won't change its meaning over Kripke MTSs. In Section 4, we first present two corresponding linear transformations of models and formulas: a transformation of a Kripke MTS into two DLTSs, and a transformation of a modal mu-calculus formula ϕ into a modal mu-calculus formula $T(\phi^+)$; both transformations extend the signature. Second, we describe a linear transformation that turns a DLTS into a Kripke structure with extended/collapsed signature and a modal mu-calculus formula ϕ into

a modal mu-calculus formula $K(\phi)$. We prove that these transformations, and therefore their compositions, preserve meaning. In Section 5, we analyze the time complexity of model checking the Kripke structures constructed in this manner; it is no greater than the time complexity of model checking Kripke structures whose size equals that of the original Kripke MTS. Section 6 discusses, for sake of illustration, how one can or cannot separate fairness from the actual model check. Section 7 discusses related work and Section 8 concludes.

2 Abstraction-based model-checking using Kripke MTSs

We begin with defining the models of interest.

Definition 1 (Doubly labeled transition systems and Kripke MTSs).

1. *A* doubly labeled transition system *[17] (DLTS)* $\mathcal{L}$ *with signature* $(\mathtt{Act}, \mathtt{AP})$ *is a tuple* (Σ, R, L)*, where* Σ *is a set of states,* $\mathtt{Act}$ *is a (countable) set of action symbols,* $\mathtt{AP}$ *is a (countable) set of atomic propositions,* R *is a transition relation with* $R \subseteq \Sigma \times \mathtt{Act} \times \Sigma$*, and* L *is a labeling function* $L\colon \Sigma \to \mathcal{P}(\mathtt{AP})$*.*
2. *A* Kripke modal transition system *[23] (Kripke MTS) with signature* $(\mathtt{Act}, \mathtt{AP})$ *is a pair* $(\mathcal{M}^{\mathrm{a}}, \mathcal{M}^{\mathrm{c}})$ *of DLTSs* $\mathcal{M}^{\mathrm{a}} = (\Sigma, R^{\mathrm{a}}, L^{\mathrm{a}})$ *and* $\mathcal{M}^{\mathrm{c}} = (\Sigma, R^{\mathrm{c}}, L^{\mathrm{c}})$ *with signature* $(\mathtt{Act}, \mathtt{AP})$ *such that* $R^{\mathrm{a}} \subseteq R^{\mathrm{c}}$ *and* $L^{\mathrm{a}}(s) \subseteq L^{\mathrm{c}}(s)$ *for all* $s \in \Sigma$*.*

It is useful to think of $\mathcal{M}^{\mathrm{a}}$ as the part of a specification that *asserts* state properties and behavior as necessary aspects of a modeled artifact, whereas $\mathcal{M}^{\mathrm{c}}$ expresses which state properties and what behavior are *consistent* (i.e. possible) with respect to the modeled artifact. E.g. in [23] Kripke MTSs are natural abstractions of a program's heap structure and in [22] they serve as abstractions of program statements as predicate transformers. As property logic for Kripke MTSs we choose, parametric in a signature $(\mathtt{Act}, \mathtt{AP})$,

$$\phi ::= \bot \mid p \mid Z \mid \neg\phi \mid \phi \wedge \phi \mid (\exists\alpha)\phi \mid \mu Z.\phi, \tag{1}$$

where $p \in \mathtt{AP}$, $\alpha \in \mathtt{Act}$, $Z \in \mathrm{var}$ for a countable set of recursion variables var, and all ϕ are formally monotone in $\mu Z.\phi$. We assume the standard embedding of $\mathtt{Act}$-CTL into (1), e.g. $\mathrm{EF}_\alpha\, p$ ("there is an α-path on which p holds eventually") translates into $\mu Z.p \vee (\exists\alpha) Z$ [4], and make liberal use of $\mathtt{Act}$-CTL connectives as abbreviations of their corresponding syntactic equivalents in (1). For $\rho = (\rho^{\mathrm{a}}, \rho^{\mathrm{c}})$ with $\rho^{\mathrm{m}}\colon \mathrm{var} \to \Sigma$ for $\mathrm{m} \in \{\mathrm{a}, \mathrm{c}\}$, we write $(\mathcal{M}, s) \models^{\mathrm{a}}_{\rho} \phi$ and $(\mathcal{M}, s) \models^{\mathrm{c}}_{\rho} \phi$ iff $s \in [\![\, \phi \,]\!]^{\mathrm{a}}_{\rho}$ and $s \in [\![\, \phi \,]\!]^{\mathrm{c}}_{\rho}$ (respectively). The denotational semantics $[\![\cdot]\!]^{\mathrm{m}}_{\cdot}$ is defined in Figure 2, where $\neg\mathrm{a} \stackrel{\mathrm{def}}{=} \mathrm{c}$, $\neg\mathrm{c} \stackrel{\mathrm{def}}{=} \mathrm{a}$, and $\mathrm{pre}^{\mathrm{m}}_{\alpha}(A) \stackrel{\mathrm{def}}{=} \{s \in \Sigma \mid \exists s' \in \Sigma\colon (s, \alpha, s') \in R^{\mathrm{m}},\ s' \in A\}$. We refer to $\mathrm{m} \in \{\mathrm{a}, \mathrm{c}\}$ as the *mode of analysis*. The semantics in Figure 2 is the standard one for DLTSs, *except* for the treatment of negation: to evaluate $\neg\phi$ in mode m, first evaluate ϕ in mode $\neg$m and then negate that result [25].

$$
\begin{aligned}
[\![\bot]\!]^{\mathrm{m}}_{\rho}\, s &\stackrel{\text{def}}{=} \{\} \\
[\![p]\!]^{\mathrm{m}}_{\rho} &\stackrel{\text{def}}{=} \{s \in \Sigma \mid p \in L^{\mathrm{m}}(s)\} \\
[\![Z]\!]^{\mathrm{m}}_{\rho} &\stackrel{\text{def}}{=} \rho^{\mathrm{m}}(Z) \\
[\![\neg\phi]\!]^{\mathrm{m}}_{\rho} &\stackrel{\text{def}}{=} \Sigma \setminus [\![\phi]\!]^{\neg\mathrm{m}}_{\rho} \\
[\![\phi_1 \wedge \phi_2]\!]^{\mathrm{m}}_{\rho} &\stackrel{\text{def}}{=} [\![\phi_1]\!]^{\mathrm{m}}_{\rho} \cap [\![\phi_2]\!]^{\mathrm{m}}_{\rho} \\
[\![(\exists\alpha)\,\phi]\!]^{\mathrm{m}}_{\rho} &\stackrel{\text{def}}{=} \mathrm{pre}^{\mathrm{m}}_{\alpha}([\![\phi]\!]^{\mathrm{m}}_{\rho}) \\
[\![\mu Z.\phi]\!]^{\mathrm{m}}_{\rho}\, s &\stackrel{\text{def}}{=} \mathrm{lfp}\, F^{\mathrm{m}};\ \text{where } F^{\mathrm{m}}(A) \stackrel{\text{def}}{=} [\![\phi]\!]^{\mathrm{m}}_{\rho^{\mathrm{m}}[Z \mapsto A]}.
\end{aligned}
$$

Fig. 2. Property semantics over Kripke MTSs [23] for mode $\mathrm{m} \in \{\mathrm{a}, \mathrm{c}\}$.

Example 1 (Laptop modes). Figure 3 shows a Kripke MTS with $\mathtt{Act} = \{*\}$ that models the modes of a laptop, where x, y, and z denote "AC powered", "battery powered", and "in suspend mode" (respectively). The labeling in the Figure means $x \in L^{\mathrm{a}}(s_0) \cap L^{\mathrm{c}}(s_0)$, $y \in L^{\mathrm{c}}(s_2) \setminus L^{\mathrm{a}}(s_2)$, and $z \in L^{\mathrm{c}}(s_1) \setminus L^{\mathrm{a}}(s_1)$. Dashed lines represent transitions in $R^{\mathrm{c}} \setminus R^{\mathrm{a}}$; solid lines denote transitions in $R^{\mathrm{a}} \cap R^{\mathrm{c}}$. The mandatory part of that model specifies the state and behavior of the laptop's AC power supply. The possible part specifies an additional power source (a battery) and a suspend mode for the machine. The property $\mathrm{AG\,EF}\, z$ — "all reachable states can reach a state in suspend mode" — is expressible in (1) as $\neg\mu Y.\neg(\mu W.z \vee ((\exists *)\,(W) \wedge (\exists *)\,\neg\bot)) \vee (\exists *)\,(Y)$. This formula is an invalid assertion[3] (we don't have $(\mathcal{M}, i) \models^{\mathrm{a}} \mathrm{AG\,EF}\, z$), but a consistent condition (we do have $(\mathcal{M}, i) \models^{\mathrm{c}} \mathrm{AG\,EF}\, z$). The evaluation of $(\mathcal{M}, i) \models^{\mathrm{a}} \mathrm{AG\,EF}\, z$ effectively checks whether all R^{c}-reachable states contain a R^{a}-path to a state s, where $z \in L^{\mathrm{a}}(s)$. The evaluation of $(\mathcal{M}, i) \models^{\mathrm{c}} \mathrm{AG\,EF}\, z$ conducts the same analysis, except that the modalities of paths are swapped.

Similarly to mixed transition systems [14, 15], the usual DeMorgan dualities are preserved by each $[\![\cdot]\!]^{\mathrm{m}}$ and $\phi \vee \neg\phi$ does not hold for $[\![\cdot]\!]^{\mathrm{a}}$ in general. However, Kripke MTSs *do* satisfy $\phi \vee \neg\phi$ for $[\![\cdot]\!]^{\mathrm{c}}$ and, equivalently, *don't* satisfy $\phi \wedge \neg\phi$ for $[\![\cdot]\!]^{\mathrm{a}}$. Although these differences seem small, our semantic approach can be transferred to interpret under-specified models $\mathcal{M}$ of software specifications and requirements, where *explicit* consistency checks ($\mathcal{M} \models^{\mathrm{c}} \phi$) and assertion checks ($\mathcal{M} \models^{\mathrm{a}} \phi$), e.g. as found in the object-modeling language Alloy [24], are vital. The soundness of abstraction-based model checking using Kripke MTSs has been shown in [23], where a co-inductive notion of refinement $(\mathcal{M}, s) \prec (\mathcal{N}, t)$ between (pointed) Kripke MTSs of the same signature is defined and proved that, for all $(\mathcal{M}, s) \prec (\mathcal{N}, t)$ and ϕ of (1) with matching signature,

[3] If convenient, we identify models $\mathcal{M}$ with pointed ones [33] $(\mathcal{M}, i)$.

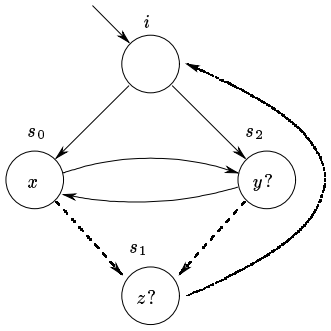

Fig. 3. A Kripke MTS modeling laptop modes.

$$\begin{array}{lcl} (\mathcal{N},t) \models^{\mathrm{a}}_{\rho} \phi & \Rightarrow & (\mathcal{M},s) \models^{\mathrm{a}}_{\rho} \phi \\ (\mathcal{M},s) \models^{\mathrm{c}}_{\rho} \phi & \Rightarrow & (\mathcal{N},t) \models^{\mathrm{c}}_{\rho} \phi. \end{array} \tag{2}$$

3 Example: Fair abstraction using Kripke MTSs

Given an Act-CTL formula ϕ and a set of fairness constraints $C = \{\psi_1, \psi_2, \ldots, \psi_n\}$ written in Act-CTL, one can express the fair semantics of ϕ with respect to C in the modal mu-calculus of alternation depth 2 [20]. Specifically, all occurrences of EX_α, EG_α, and $\mathrm{E}[\cdot\,\mathrm{U}_\alpha\,\cdot]$ are replaced by their fair versions $\mathrm{E}_C\mathrm{X}_\alpha$, $\mathrm{E}_C\mathrm{G}_\alpha$, and $\mathrm{E}_C[\cdot\,\mathrm{U}_\alpha\,\cdot]$:

$$\mathrm{E}_C\mathrm{G}_\alpha\,\phi \stackrel{\mathrm{def}}{=} \neg(\mu Z.\neg(\phi \wedge \bigwedge_{i=1}^{n} \mathrm{EX}_\alpha\,\mathrm{E}[f\,\mathrm{U}_\alpha\,\neg Z \wedge \psi_i])) \tag{3}$$

$$\mathrm{E}_C\mathrm{X}_\alpha\,\phi \stackrel{\mathrm{def}}{=} \mathrm{EX}_\alpha\,(\phi \wedge \mathrm{E}_C\mathrm{G}\neg\bot) \tag{4}$$

$$\mathrm{E}_\alpha[\phi\,\mathrm{U}_C\,\eta] \stackrel{\mathrm{def}}{=} \mathrm{E}[\phi\,\mathrm{U}_\alpha\,\eta \wedge \mathrm{E}_C\mathrm{G}\,\neg\bot]. \tag{5}$$

In that manner, ϕ is translated into a formula of the modal mu-calculus of alternation depth 2. Note that alternation depths are defined through the positive normal form of formulas [20], so changing a formula into its positive normal form will not change its alternation depth, nor its meaning over Kripke MTSs. Finally, and crucially, the implications in (2) guarantee that model checking such encodings on abstract Kripke MTSs is sound. This is needed since the normal form of (3) is $\nu Z.\phi \wedge \bigwedge_{i=1}^{n} \mathrm{EX}_\alpha\,\mathrm{E}[f\,\mathrm{U}_\alpha\,Z \wedge \psi_i]$) which combines existential (the least fixed point for $\mathrm{E}[\cdot\,\mathrm{U}.\,\cdot]$) and universal (the greatest fixed point for νZ) aspects in one property.

4 Sound abstraction using Kripke structures

For each $k \geq 0$, we transform the abstraction-based model checking problem of the alternation-depth k modal mu-calculus for Kripke MTSs with signature $(\mathtt{Act}, \mathtt{AP})$ to an alternation-depth k model-checking problem of the modal mu-calculus for Kripke structures with signature[4] $\mathtt{Act} + \mathtt{Act} + \mathtt{AP}$, where the transformations are linear in the size of models and formulas.

Definition 2 (Kripke structures). *A* Kripke structure $\mathcal{K}$ *with signature* $\mathtt{AP}$ *is a tuple* (Σ, R, L)*, where* Σ *is a set of states,* $\mathtt{AP}$ *is a (countable) set of atomic propositions,* $R \subseteq \Sigma \times \Sigma$*, and* L *is a labeling function* $L\colon \Sigma \to \mathcal{P}(\mathtt{AP})$*.*

We achieve this reduction by first reducing the model-checking problem for Kripke MTSs to two model-checking problems for DLTSs. As a property logic for DLTSs, parametric in a signature $(\mathtt{Act}, \mathtt{AP})$, we use the modal mu-calculus augmented with the duals of the clauses in (1):

$$\phi ::= \bot \mid \top \mid Z \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid (\exists\alpha)\phi \mid (\forall\alpha)\phi \mid \mu Z.\phi \mid \nu Z.\phi, \qquad (6)$$

where $p \in \mathtt{AP}$, $\alpha \in \mathtt{Act}$, $Z \in \mathrm{var}$, and ϕ is formally monotone in $\mu Z.\phi$ and $\nu Z.\phi$. The semantics $[\![\ \phi\ ]\!]_\rho$ over DLTS is the standard one, e.g. see [4], and we write $(\mathcal{L}, s) \models_\rho \phi$ for $s \in [\![\ \phi\ ]\!]_\rho$. There is a fairly rich literature on conversions of one kind of non-deterministic model into another, e.g. transforming Moore machines into Kripke structures [28], and Kripke structures into Büchi automata (see e.g. [36]) or DLTSs [17]. The significance of the latter transformation is that it maps one kind of observational equivalence (e.g. stuttering equivalence [5]) into another (e.g. branching bisimulation [35]). These equivalences have logical characterizations, but our intent of such transformations is more specific in that we seek to preserve meanings for all model checks $\mathcal{M} \models \phi$ of one logic and class of models by transforming formulas and models, $\mathcal{M} \mapsto \mathcal{M}'$ and $\phi \mapsto \phi'$, such that the model check $\mathcal{M} \models \phi$ is equivalent to the check $\mathcal{M}' \models \phi'$, for all $\mathcal{M}$ and ϕ. We proceed in two stages.

Stage #1. In a straightforward adaptation of a transformation of modal transition systems [22], we transform a Kripke MTS $\mathcal{M}$ with signature $(\mathtt{Act}, \mathtt{AP})$ into two DLTSs $\mathcal{M}^{\mathrm{p}}$ and $\mathcal{M}^{\mathrm{o}}$ of an extended signature $(\overline{\mathtt{Act}}, \mathtt{AP})$. By construction, checking ϕ in mode a and c on the Kripke MTS $\mathcal{M}$ is equivalent to checking a transformed formula $T(\phi^+)$ on the DLTSs $\mathcal{M}^{\mathrm{p}}$ and $\mathcal{M}^{\mathrm{o}}$ (respectively). This had already been done for MTSs and LTSs in [22]. Given a Kripke MTS $\mathcal{M} = ((\Sigma, R^{\mathrm{a}}, L^{\mathrm{a}}), (\Sigma, R^{\mathrm{c}}, L^{\mathrm{c}}))$, with signature $(\mathtt{Act}, \mathtt{AP})$, we define two DLTSs $\mathcal{M}^{\mathrm{p}} \stackrel{\mathrm{def}}{=} (\Sigma, R^{\mathrm{p}}, L^{\mathrm{a}})$ and $\mathcal{M}^{\mathrm{o}} \stackrel{\mathrm{def}}{=} (\Sigma, R^{\mathrm{o}}, L^{\mathrm{c}})$ with signature $(\overline{\mathtt{Act}}, \mathtt{AP})$ — representing the *pessimistic* and *optimistic* interpretations [7] of $\mathcal{M}$ (respectively):

$$\overline{\mathtt{Act}} \stackrel{\mathrm{def}}{=} \{\alpha_\forall \mid \alpha \in \mathtt{Act}\} \cup \{\alpha_\exists \mid \alpha \in \mathtt{Act}\} \qquad (7)$$

[4] We write $+$ to denote disjoint union of sets.

$$R^{\mathrm{p}} \stackrel{\text{def}}{=} \{(s,\alpha_\forall,s') \mid (s,\alpha,s') \in R^{\mathrm{c}}\} \cup \{(s,\alpha_\exists,s') \mid (s,\alpha,s') \in R^{\mathrm{a}}\} \quad (8)$$
$$R^{\mathrm{o}} \stackrel{\text{def}}{=} \{(s,\alpha_\forall,s') \mid (s,\alpha,s') \in R^{\mathrm{a}}\} \cup \{(s,\alpha_\exists,s') \mid (s,\alpha,s') \in R^{\mathrm{c}}\}.$$

We transform all formulas ϕ of (1) with signature $(\mathtt{Act},\mathtt{AP})$ to formulas ϕ^+ of (6) with the same signature by applying the classical rewrite rules

$$\neg\neg\phi \rightsquigarrow \phi \qquad \neg(\phi_1 \wedge \phi_2) \rightsquigarrow (\neg\phi_1) \vee (\neg\phi_2)$$
$$\neg((\exists\alpha)\phi) \rightsquigarrow (\forall\alpha)(\neg\phi) \qquad \neg(\mu Z.\phi) \rightsquigarrow \nu Z.(\neg\phi).$$

Finally, $T(\phi^+)$ is a formulas of (6) with signature $(\overline{\mathtt{Act}},\mathtt{AP})$ and is computed from ϕ^+ as follows: for all $\alpha \in \mathtt{Act}$, we replace all occurrences of $(\forall\alpha)$ in ϕ^+ by $(\forall\alpha_\forall)$ and all occurrences of $(\exists\alpha)$ in ϕ^+ by $(\exists\alpha_\exists)$.

Theorem 1 (Correctness of first reduction [22]). *Given a Kripke MTS $\mathcal{M}$ with signature $(\mathtt{Act},\mathtt{AP})$ and any ϕ of (1) with matching signature, let $T(\phi^+)$, $\mathcal{M}^{\mathrm{p}}$, and $\mathcal{M}^{\mathrm{o}}$ be the formula and the two DLTSs (respectively) as defined above. For any state $s \in \Sigma$, we then have*

$$(\mathcal{M},s) \models^{\mathrm{a}}_{\rho} \phi \quad \textit{iff} \quad (\mathcal{M}^{\mathrm{p}},s) \models_{\rho} T(\phi^+) \quad (9)$$
$$(\mathcal{M},s) \models^{\mathrm{c}}_{\rho} \phi \quad \textit{iff} \quad (\mathcal{M}^{\mathrm{o}},s) \models_{\rho} T(\phi^+). \quad (10)$$

Stage #2. We parametrically define a transformation $\phi \mapsto K(\phi)$ of formulas ϕ in (6) with signature $(\mathtt{Act},\mathtt{AP})$ into formulas of

$$\phi ::= \bot \mid \top \mid Z \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathrm{EX}\,\phi \mid \mathrm{AX}\,\phi \mid \mu Z.\phi \mid \nu Z.\phi, \quad (11)$$

with signature $\mathtt{Act} \cup \mathtt{AP}$, the range of p:

$$K(\bot) \stackrel{\text{def}}{=} \bot \qquad K(\top) \stackrel{\text{def}}{=} \top$$
$$K(Z) \stackrel{\text{def}}{=} Z \qquad K(p) \stackrel{\text{def}}{=} p$$
$$K(\neg\phi) \stackrel{\text{def}}{=} \neg K(\phi)$$
$$K(\phi_1 \wedge \phi_2) \stackrel{\text{def}}{=} K(\phi_1) \wedge K(\phi_2) \qquad K(\phi_1 \vee \phi_2) \stackrel{\text{def}}{=} K(\phi_1) \vee K(\phi_2)$$
$$K((\exists\alpha)\phi) \stackrel{\text{def}}{=} \mathrm{EX}\,(K(\phi) \wedge \alpha) \qquad K((\forall\alpha)\phi) \stackrel{\text{def}}{=} \mathrm{AX}\,(\neg\alpha \vee K(\phi))$$
$$K(\mu Z.\phi) \stackrel{\text{def}}{=} \mu Z.K(\phi) \qquad K(\nu Z.\phi) \stackrel{\text{def}}{=} \nu Z.K(\phi).$$

Note that the transformations for $(\exists\alpha)\phi$ and $(\forall\alpha)\phi$ are the only clauses that change subformulas and cause the signature extension/collapse to $\mathtt{Act} \cup \mathtt{AP}$. In particular, $\phi \mapsto K(\phi)$ does not change ϕ's recursive structure, so ϕ and $K(\phi)$ have the same alternation depth. Next, we transform DLTSs into Kripke structures.

Definition 3 (Induced Kripke structure). *Given a DLTS* $\mathcal{L} = (\Sigma, R, L)$ *with signature* $(\mathtt{Act}, \mathtt{AP})$*, we define a Kripke structure* $\mathrm{K}[\mathcal{L}] = (\Sigma \times \mathtt{Act}, \bar{R}, \bar{L})$ *with signature* $\mathtt{Act} \cup \mathtt{AP}$ *where*

$$\bar{R} \stackrel{\text{def}}{=} \{((s,\alpha),(s',\beta)) \mid \alpha \in \mathtt{Act},\ (s,\beta,s') \in R\} \tag{12}$$

$$\bar{L}(s,\alpha) \stackrel{\text{def}}{=} L(s) \cup \{\alpha\}. \tag{13}$$

The semantics of (11) over Kripke structures with signature $\mathtt{AP} \cup \mathtt{Act}$ is the standard one — e.g. see [4] — and we also denote it with $[\![\ \phi\ ]\!]_\rho$ since the context will determine the logic and model. As usual, we write $(\mathcal{K}, s) \models_\rho \phi$ for $s \in [\![\ \phi\ ]\!]_\rho$. Finally, we prove that the original model check of ϕ in the DTLS $\mathcal{L}$ is captured by the model check of $K(\phi)$ in the induced Kripke structure $\mathrm{K}[\mathcal{L}]$.

Theorem 2 (Correctness of second reduction). *Let* $\mathcal{L}$ *be a DLTS with signature* $(\mathtt{Act}, \mathtt{AP})$ *such that* $\mathtt{AP} \cap \mathtt{Act} = \{\}$*. For any* $\rho\colon \mathrm{var} \to \mathcal{P}(\Sigma)$ *define* $\rho_\mathcal{K}\colon \mathrm{var} \to \mathcal{P}(\Sigma \times \mathtt{Act})$ *by* $\rho_\mathcal{K}(Z) \stackrel{\text{def}}{=} \rho(Z) \times \mathtt{Act}$*. For any* ϕ *from (6) with signature* $(\mathtt{Act}, \mathtt{AP})$*, any* $s \in \Sigma$*, and any* ρ *as above, we have*

$$[\![\ \phi\ ]\!]_\rho \times \mathtt{Act} = [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}}. \tag{14}$$

Proof. The cases $\bot$, $\top$ are immediate and $\wedge$ and $\vee$ follow by induction. The case Z follows from the definition of $\rho_\mathcal{K}$. The case p holds due to $\mathtt{Act} \cap \mathtt{AP} = \{\}$.

- We have $[\![\ \neg\phi\ ]\!]_\rho \times \mathtt{Act} = (\Sigma \setminus [\![\ \phi\ ]\!]_\rho) \times \mathtt{Act} = (\Sigma \times \mathtt{Act}) \setminus ([\![\ \phi\ ]\!]_{\rho_\mathcal{K}} \times \mathtt{Act}) = (\Sigma \times \mathtt{Act}) \setminus [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}} = [\![\ \neg K(\phi)\ ]\!]_{\rho_\mathcal{K}} = [\![\ K(\neg\phi)\ ]\!]_{\rho_\mathcal{K}}$.
- Let $\alpha \in \mathtt{Act}$. Given $s \in [\![\ (\exists\beta)\phi\ ]\!]_\rho$, there exists some $s' \in \Sigma$ with $(s,\beta,s') \in R$ and $s' \in [\![\ \phi\ ]\!]_\rho$. By induction, $(s',\beta) \in [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}}$. By definition, $(s',\beta) \in [\![\ \beta\ ]\!]_{\rho_\mathcal{K}}$. Thus, $(s',\beta) \in [\![\ K(\phi) \wedge \beta\ ]\!]_{\rho_\mathcal{K}}$. But $(s,\beta,s') \in R$ implies $((s,\alpha),(s',\beta)) \in \bar{R}$ and so $(s,\alpha) \in [\![\ \mathrm{EX}\,(K(\phi) \wedge \beta)\ ]\!]_{\rho_\mathcal{K}} = [\![\ K((\exists\beta)\phi)\ ]\!]_{\rho_\mathcal{K}}$. Conversely, let $(s,\alpha) \in [\![\ K((\exists\beta)\phi)\ ]\!]_{\rho_\mathcal{K}} = [\![\ \mathrm{EX}\,(K(\phi) \wedge \beta)\ ]\!]_{\rho_\mathcal{K}}$. Then there exists some $((s,\alpha),(s',\gamma)) \in \bar{R}$ with $(s',\gamma) \in [\![\ K(\phi) \wedge \beta\ ]\!]_{\rho_\mathcal{K}}$. In particular, $(s',\gamma) \in [\![\ \beta\ ]\!]_{\rho_\mathcal{K}}$, which implies $\gamma = \beta$. But then $(s,\beta,s') \in R$ follows. By induction, $s' \in [\![\ \phi\ ]\!]_\rho$. Therefore, $s \in [\![\ (\exists\beta)\phi\ ]\!]_\rho$.
- Let $\alpha \in \mathtt{Act}$. We have $s \in [\![\ (\forall\beta)\phi\ ]\!]_\rho$ iff for all $s' \in \Sigma$, $(s,\beta,s') \in R \Rightarrow s' \in [\![\ \phi\ ]\!]_\rho$ iff for all $s' \in \Sigma$, $((s,\alpha),(s',\beta)) \in \bar{R} \Rightarrow (s',\beta) \in [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}}$ iff for all $s' \in \Sigma$ and for all $\gamma \in \mathtt{Act}$, $((s,\alpha),(s',\gamma)) \in \bar{R}\ \&\ \gamma = \beta \Rightarrow (s',\beta) \in [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}}$ iff for all $s' \in \Sigma$ and for all $\gamma \in \mathtt{Act}$, $((s,\alpha),(s',\gamma)) \in \bar{R} \Rightarrow (s',\gamma) \in [\![\ \neg\beta \vee K(\phi)\ ]\!]_{\rho_\mathcal{K}}$ iff $(s,\alpha) \in [\![\ \mathrm{AX}\,(\neg\beta \vee K(\phi))\ ]\!]_{\rho_\mathcal{K}} = [\![\ K((\forall\beta)\phi)\ ]\!]_{\rho_\mathcal{K}}$.
- As for μZ and νZ, consider

  $$F\colon \mathcal{P}(\Sigma) \to \mathcal{P}(\Sigma) \qquad F(A) \stackrel{\text{def}}{=} [\![\ \phi\ ]\!]_{\rho[Z \mapsto A]}$$
  $$G\colon \mathcal{P}(\Sigma \times \mathtt{Act}) \to \mathcal{P}(\Sigma \times \mathtt{Act}) \qquad G(B) \stackrel{\text{def}}{=} [\![\ K(\phi)\ ]\!]_{\rho_\mathcal{K}[Z \mapsto B]}.$$

  By induction, $F(A) \times \mathtt{Act} = G(A \times \mathtt{Act})$ for all $A \subseteq \Sigma$, since the environments $(\rho[Z \mapsto A])_\mathcal{K}$ and $\rho_\mathcal{K}[Z \mapsto A \times \mathtt{Act}]$ are equal. But $\mu Z.\phi/\nu Z.\phi$ and

$\mu Z.K(\phi)/\nu Z.K(\phi)$ are the least/greatest fixed points of F and G (respectively), the function $A \mapsto A \times \mathtt{Act} : \mathcal{P}(\Sigma) \to \mathcal{P}(\Sigma \times \mathtt{Act})$ preserves all unions and intersections, and all fixed-point approximations for G are of the form $A \times \mathtt{Act}$ for some $A \subseteq \Sigma$. □

Of course, the model check on the right-hand side of (14) is performed over a Kripke structure of signature $\mathtt{Act} \cup \mathtt{AP}$. We can combine our two constructions to reduce model-checking a Kripke MTS $\mathcal{M} = (\mathcal{M}^{\mathrm{a}}, \mathcal{M}^{\mathrm{c}})$ with signature $(\mathtt{Act}, \mathtt{AP})$ to model-checking a Kripke structure with signature $\overline{\mathtt{Act}} \cup \mathtt{AP}$. For $\models^{\mathrm{a}}$, we model check $\mathrm{K}[\mathcal{M}^{\mathrm{o}}]$, for $\models^{\mathrm{c}}$, we model check $\mathrm{K}[\mathcal{M}^{\mathrm{p}}]$.

Corollary 1 (Composite model-checking reduction). *Let $\mathcal{M} = (\mathcal{M}^{\mathrm{a}}, \mathcal{M}^{\mathrm{c}})$ be a Kripke MTS with signature $(\mathtt{Act}, \mathtt{AP})$. For $\mathcal{M}^{\mathrm{p}}$ and $\mathcal{M}^{\mathrm{o}}$ as above, we have*

$$(\mathcal{M}, s) \models^{\mathrm{a}}_{\rho} \phi \qquad \textit{iff} \qquad (\mathrm{K}[\mathcal{M}^{\mathrm{p}}], (s, \alpha)) \models_{\rho_{\mathcal{K}}} K(T(\phi^{+})) \tag{15}$$

$$(\mathcal{M}, s) \models^{\mathrm{c}}_{\rho} \phi \qquad \textit{iff} \qquad (\mathrm{K}[\mathcal{M}^{\mathrm{o}}], (s, \alpha)) \models_{\rho_{\mathcal{K}}} K(T(\phi^{+})) \tag{16}$$

for all ϕ of (1), ρ: var $\to \mathcal{P}(\Sigma)$, $s \in \Sigma$, and $\alpha \in \mathtt{Act}$.

Example 2 (Computing $K(T(\phi^{+}))$). Consider $\mathtt{Act} = \{*\}$ and the $\mathtt{Act}$-CTL formula $\mathrm{AF}_{*}\, p$ under a sole simple fairness constraint $C = \{q\}$.

1. We can express $\mathrm{AF}_{*}\, p$ in (1) as $\neg \mathrm{EG}_{*}\, \neg p$.
2. We convert the EG_{*} sub-formula to its fair version: $\phi \stackrel{\text{def}}{=} \neg \mathrm{E}_{C} \mathrm{G}_{*}\, \neg p = \neg(\nu Z.\neg p \wedge \mathrm{EX}_{*}\, (\mathrm{E}[\neg p\, \mathrm{U}_{*}\, Z \wedge q]))$.
3. We compute the positive normal form

$$\phi^{+} \;=\; \mu Z.p \vee \mathrm{AX}_{*}\, (\nu Y.(Z \vee \neg q) \wedge (p \vee \mathrm{AX}_{*}\, Y)). \tag{17}$$

4. We change the actions $*$ attached to quantifiers to compute

$$T(\phi^{+}) \;=\; \mu Z.p \vee \mathrm{AX}_{*_{\forall}}\, (\nu Y.(Z \vee \neg q) \wedge (p \vee \mathrm{AX}_{*_{\forall}}\, Y)). \tag{18}$$

5. Applying K to $T(\phi^{+})$ does not do anything material in this example as its input formula mentions one action only. Note that this is not so in general, even for $\mathtt{Act} = \{*\}$.

5 Complexity of model-checking reduction

We measure the size of the Kripke structures $\mathrm{K}[\mathcal{M}^{\mathrm{p}}]$ and $\mathrm{K}[\mathcal{M}^{\mathrm{o}}]$ in terms of the size of the Kripke MTS $\mathcal{M}$, showing the there is no significant increase in the size of models. Similarly, the transformation of formulas ϕ into $K(T(\phi^{+}))$ is linear.

Definition 4 (Model complexity). *Let $\mathcal{L} = (\Sigma, R, L)$ be a DLTS with signature $(\mathtt{Act}, \mathtt{AP})$. The* model complexity *[20] of $\mathcal{L}$ is $|\mathcal{L}| \stackrel{\text{def}}{=} |\Sigma| + |R|$, where $|R| \stackrel{\text{def}}{=} \sum_{\alpha \in \mathtt{Act}} |\{(s, s') \mid (s, \alpha, s') \in R\}|$. For a Kripke MTS $\mathcal{M} = (\mathcal{M}^{\mathrm{a}}, \mathcal{M}^{\mathrm{c}})$ we define its model complexity as $|\mathcal{M}| \stackrel{\text{def}}{=} |\mathcal{M}^{\mathrm{a}}| + |\mathcal{M}^{\mathrm{c}}|$.*

Theorem 3 (Model-checking complexity). *Let $\mathcal{M} = (\mathcal{M}^{\mathrm{a}}, \mathcal{M}^{\mathrm{c}})$ be a Kripke MTS with finite signature $(\mathtt{Act}, \mathtt{AP})$.*

1. *Let $\mathcal{L}$ be either $\mathrm{K}[\mathcal{M}^{\mathrm{p}}]$ or $\mathrm{K}[\mathcal{M}^{\mathrm{o}}]$. Then*

$$|\mathcal{L}| = |\mathtt{Act}| \cdot (|\Sigma| + 2 \cdot (|R^{\mathrm{a}}| + |R^{\mathrm{c}}|)) \leq 2 \cdot |\mathtt{Act}| \cdot |\mathcal{M}|. \tag{19}$$

   If ϕ of (1) has alternation depth k, then the time complexity for model checking $K(T(\phi^+))$ over $\mathcal{L}$ is in $O(|\phi| \cdot |\mathtt{Act}|^{k+1} \cdot |\mathcal{M}|^{k+1})$.[5]
2. *If the Kripke MTS $\mathcal{M}$ has only one action type, then*

$$|\mathcal{L}| = |\Sigma| + 2 \cdot (|R^{\mathrm{a}}| + |R^{\mathrm{c}}|) \leq 2 \cdot |\mathcal{M}| \tag{20}$$

   and the time complexity for model checking such a ϕ over $\mathcal{L}$ is in $O(|\phi| \cdot |\mathcal{M}|^{k+1})$.

Proof. The computation of model complexities is straightforward. As for the time complexities, they follow from the model complexities, the complexity bound given in [20], and the fact that $\phi \mapsto T(\phi^+) \mapsto K(T(\phi^+))$ is a sequence of linear transformations that each preserve the alternation depth of formulas. □

We emphasize that the time complexity in item 2 is identical to the one obtained if $\mathcal{M}$ were a Kripke structure already, i.e. if $\mathcal{M}^{\mathrm{a}}$ equaled $\mathcal{M}^{\mathrm{c}}$ in that case [20]. Of course, our model-checking reduction allows the use of *any* efficient model checking algorithms for Kripke structures — be they established tableau methods [4] or more recent advances in automata-theoretic approaches to model checking, such as hesitant alternating automata [36] for the CTL* fragment of (11).

6 Example: Separating fairness algorithmically

The modal mu-calculus encoding for model-checking fair CTL has more efficient algorithms that separate the fairness constraints from the CTL formula ϕ to be checked [8, 9]. These techniques can be applied to a Kripke MTS $\mathcal{M}$ of signature $(\{*\}, \mathtt{AP})$: compute the fair maximal connected components of $\mathcal{M}^{\mathrm{a}}$ and $\mathcal{M}^{\mathrm{c}}$, or adapt the more space efficient methods of [11], and then restrict the semantics $\models^{\mathrm{a}}$ and $\models^{\mathrm{c}}$ to those states that lead into a fair maximally connected component. Unfortunately, these two model checks are mutually dependent and can therefore not be emulated in standard tools per se. It would be of interest to see whether our model-checking reduction can achieve a similar separation of concerns. Alas, the definitions in (8) "mix" state transitions of $\mathcal{M}^{\mathrm{a}}$ with state transitions of $\mathcal{M}^{\mathrm{c}}$, preventing a direct detection of fair "assertion-paths" and "consistency-paths" in the model $\mathrm{K}[\mathcal{M}^{\mathrm{a}}]$ or $\mathrm{K}[\mathcal{M}^{\mathrm{c}}]$ in isolation. Thus, the reduction of three-valued to two-valued model checking not only results in a loss of precision in the interpretation of conjunction, as discussed in [7], it may also require new tools

[5] Or $O(|\phi| \cdot |\mathcal{M}|^{k+1})$ is we consider $|\mathtt{Act}|$ to be constant.

to maintain the expressiveness needed in practice, e.g. for property verification under fairness assumptions.

Although our use of alternation-depth 2 model-checking introduces a computational penalty, there are good reasons beyond fairness for wanting to use properties of that fragment, an example being "for all paths, if the device is reset then there is some path on which it is eventually in its initial mode" which is not expressible as an alternation-depth 1 formula [20]. At the same time, it is unclear how severe that penalty really is, considering the progress made in automata-theoretic approaches to model checking modal mu-calculus formulas with hesitant alternating automata; e.g. for the alternation depth 1 fragment [3] and CTL* [36]. Since these approaches take a Kripke structure and a formula as input, our reduction enables the sound use of these tools for abstraction-based model checking of Kripke MTSs.

7 Related work

Bruns & Godefroid [7] pioneered the reduction of three-valued model-checking problems to two-valued ones for partial Kripke structures [6], which are three-valued versions of Kripke structures $\mathcal{K} = (\Sigma, R, L)$, where the labeling function L has type $L\colon \Sigma \times \mathtt{AP} \to \{\mathsf{false}, \mathsf{true}, \bot\}$. Thus, atomic propositions at states are either false, true, or $\bot$ (undetermined). They also transform such a structure into two Kripke structures and a modal mu-calculus formula into one in positive normal form. Their model-checking reduction is also sound and complete and does not increase the size of the models, nor the cost of the model check.

Our reduction proceeds in two stages, one of which is an adaptation of a reduction of model-checking modal transition systems to model-checking labeled transition systems [22]. In that paper it was shown that abstraction-based model checking using modal transition systems incurs no additional cost or complexity over abstraction-based model checking of labeled transition systems.

Dams et al. [16] study fair CTL* over mixed transition systems [14, 15], where transitions and propositions came in two flavors: free and constrained ones. Their fairness assumptions are boolean combinations of "infinitely often L", where L is a literal. To gain efficiency offered by some model-checking tools, they separate the fairness assumptions from the CTL formula ϕ to be checked. They convert ϕ into positive normal form and annotate ϕ on path quantifiers and atomic propositions. These annotations guide the satisfaction relation in its choice of free or constrained fair paths. The soundness of that approach for fair CTL* over mixed transition systems is then proved.

The semantics of $\vee$ for $\models^{\mathrm{a}}$ is an under-approximation, as the model check $s \models^{\mathrm{a}} p \vee \neg p$ with $p \in L^{\mathrm{c}}(s) \setminus L^{\mathrm{a}}(s)$ shows. Dually, the semantics of $\wedge$ for $\models c$ is an over-approximation, considering the model check $s \models^{\mathrm{c}} p \wedge \neg p$ with $p \in L^{\mathrm{c}}(s) \setminus L^{\mathrm{a}}(s)$. Generalized model checking [7] eliminates such imprecision for

partial Kripke structures [6] — a special class of Kripke MTSs (see Figure 1) — but increases the model-checking complexity.[6]

There exist linear-time temporal logics whose expressiveness exceeds that of LTL. For example, Intel developed a model checker for a linear-time temporal logic FTL [1] whose expressiveness supports a limited form of past tense modalities, subsumes ω-regular expressions (achieved through several redundant mechanisms), and contains a variety of syntactic support for hardware verification (e.g. multiple clocks, reset signals, and temporal connectives over time windows).

8 Conclusion

In [23], a model-checking framework based on Kripke modal transition systems was presented and shown that it allows sound abstraction-based model checking for the entire modal mu-calculus. In [22], it was demonstrated that abstract Kripke modal transition systems can be computed with a cost no greater than the computation of standard abstract doubly labeled transition systems. In this paper, we presented a transformation of modal mu-calculus formulas and Kripke modal transition systems into modal mu-calculus formulas and Kripke structures of an extended/collapsed signature such that this transformation is linear in the size of formulas and models and that it preserves the meaning of model checks. Specifically, for each mode of analysis $\mathrm{m} \in \{\mathrm{a}, \mathrm{c}\}$ a different Kripke structure is computed, whereas the transformed formula is the same in each mode. Since these transformations preserve the alternation depth of formulas, as well as the CTL and CTL* fragments, this model-checking reduction allows the instrumented use of efficient Kripke structure model checkers for the model checking of Kripke modal transition systems.

Acknowledgments

We wish to thank Patrice Godefroid and Radha Jagadeesan for inspiring discussions and most helpful comments. The anonymous referees are thanked for their constructive feedback.

References

1. R. Armoni, L. Fix, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, A. Tiemeyer, E. Singerman, and M. Y. Vardi. The ForSpec temporal language: A new temporal property-specification language. Submitted, 2001.

[6] If it turns out that Kripke MTSs can be translated to partial Kripke structures such that refinements are preserved and reflected, then the generalized model checking of [7] can be applied to Kripke MTSs as well.

2. T. Ball, A. Podelski, and S. K. Rajamani. Boolean and Cartesian Abstraction for Model Checking C Programs. In T. Margaria and W. Yi, editors, *Proceedings of TACAS'2001*, volume 2031 of *LNCS*, pages 268–283, Genova, Italy, April 2001. Springer Verlag.
3. O. Bernholtz, M. Vardi, and P. Wolper. An Automata-Theoretic Approach to Branching-Time Model-Checking. In *6th Int'l Conference on Computer Aided Verification (CAV'94)*, volume 818 of *Lecture Notes in Computer Science*, pages 142–155. Springer Verlag, 1994.
4. J. C. Bradfield. *Verifying Temporal Properties Of Systems.* Birkhäuser, Boston, Mass., 1991.
5. M. C. Browne, E. M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59(1–2):115–131, 1988.
6. G. Bruns and P. Godefroid. Model Checking Partial State Spaces with 3-Valued Temporal Logics. In *Proceedings of the 11th Conference on Computer Aided Verification*, volume 1633 of *Lecture Notes in Computer Science*, pages 274–287. Springer Verlag, July 1999.
7. G. Bruns and P. Godefroid. Generalized Model Checking: Reasoning about Partial State Spaces. In *Proceedings of CONCUR'2000 (11th International Conference on Concurrency Theory)*, volume 1877 of *Lecture Notes in Computer Science*, pages 168–182. Springer Verlag, August 2000.
8. J. R. Burch, E. M. Clarke, D. L. Dill K. L. McMillan, and J. Hwang. Symbolic model checking: 10^{20} states and beyond. Proceedings of the Fifth Annual Symposium on Logic in Computer Science, June 1990.
9. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, January 2000.
10. E.M. Clarke, O. Grumberg, and D.E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.
11. C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient Algorithms for the Verification of Temporal Properties. *Formal Methods in System Design*, 1(275–288), 1992.
12. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs. In *Proc. 4th ACM Symp. on Principles of Programming Languages*, pages 238–252. ACM Press, 1977.
13. P. Cousot and R. Cousot. Temporal abstract interpretation. In *Conference Record of the 27th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 12–25, Boston, Mass., January 2000. ACM Press, New York, NY.
14. D. Dams. *Abstract interpretation and partition refinement for model checking.* PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.
15. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19(2):253–291, 1997.
16. D. Dams, R. Gerth, and O. Grumberg. Fair Model Checking Of Abstractions. In M. Leuschel, A. Podelski, C.R. Ramakrishnan, and U. Ultes-Nitsche, editors, *Proceedings of the Workshop on Verification and Computational Logic (VCL'2000)*, DSSE-TR-2000-6. University of Southhampton, July 2000.
17. R. de Nicola and F. Vaandrager. Three Logics for Branching Bisimulation. *Journal of the Association of Computing Machinery*, 42(2):458–487, March 1995.
18. M. B. Dwyer, J. Hatcliff, R. Joehanes, S. Laubach, C. S. Pasareanu, Robby, W. Visser, and H. Zheng. Tool-supported Program Abstraction for Finite-state

Verification. In *Proceedings of the 23nd Intl' Conference on Software Engineering*, pages 177–187. ACM Press, May 2001.
19. M. B. Dwyer and D. A. Schmidt. Limiting State Explosion with Filter-Based Refinement. In *Proceedings of the ILPS'97 Workshop on Verification, Model Checking, and Abstraction*, 1997.
20. E. A. Emerson and C. L. Lei. Efficient Model Checking in Fragments of the Mu-calculus. In *Proc. of the First Int'l IEEE Symposium on Logic in Computer Science (LICS'86)*, pages 267–278, Cambridge, Mass., June 1986. IEEE Press.
21. N. Francez. *Fairness*. Texts and Monographs in Computer Science. Springer Verlag, 1986.
22. P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based Model Checking using Modal Transition Systems. In *Proceedings of the International Conference on Theory and Practice of Concurrency*, Lecture Notes in Computer Science, pages 426–440. Springer Verlag, August 2001.
23. M. Huth, R. Jagadeesan, and D. Schmidt. Modal transition systems: a foundation for three-valued program analysis. In Sands D., editor, *Proceedings of the European Symposium on Programming (ESOP'2001)*, pages 155–169. Springer Verlag, April 2001.
24. D. Jackson. Alloy: A Lightweight Object Modelling Language. Technical Report TR-797, Laboratory of Computer Science, Massachusetts Institute of Technology, 28 July 2000.
25. P. Kelb. Model checking and abstraction: a framework preserving both truth and failure information. Technical Report OFFIS, University of Oldenburg, Germany, 1994.
26. K.L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.
27. K. G. Larsen and B. Thomsen. A Modal Process Logic. In *Third Annual Symposium on Logic in Computer Science*, pages 203–210. IEEE Computer Society Press, 1988.
28. D. E. Long. *Model Checking, Abstraction, and Compositional Verification*. PhD thesis, Carnegie Mellon University, School of Computer Science, July 1993.
29. R. Milner. An algebraic definition of simulation between programs. In *2nd International Joint Conference on Artificial Intelligence*, pages 481–489, London, United Kingdom, 1971. British Computer Society.
30. R. Milner. A modal characterisation of observable machine behaviours. In G. Astesiano and C. Böhm, editors, *CAAP '81*, volume 112 of *Lecture Notes in Computer Science*, pages 25–34. Springer Verlag, 1981.
31. M. Sagiv, T. Reps, and R. Wilhelm. Parametric Shape Analysis via 3-Valued Logic. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of programming languages*, pages 105–118, January 20-22, San Antonio, Texas 1999.
32. H. Saidi and N. Shankar. Abstract and model check while you prove. In *Proc. of the 11th Conference on Computer-Aided Verification*, number 1633 in Lecture Notes in Computer Science, pages 443–454. Springer, 1999.
33. D. A. Schmidt. Binary relations for abstraction and refinement. *Elsevier Electronic Notes in Computer Science*, November 1999. Workshop on Refinement and Abstraction, Osaka, Japan. To appear.
34. David A. Schmidt. From Trace Sets to Modal Transition Systems. Submitted for publication, July 2001.
35. R. J. van Glabbeek and W. P. Weijland. Branching Time and Abstraction in Bisimulation Semantics. *Journal of the ACM*, 43(3):555–600, May 1996.
36. W. Visser and H. Barringer. Practical CTL* Model Checking — Should SPIN be Extended? *Software Tools for Technology Transfer*, 2(4), 2000.