Multi-Valued Model Checking via Classical Model Checking

Arie Gurfinkel and Marsha Chechik

Department of Computer Science, University of Toronto,
Toronto, ON M5S 3G4, Canada.
Email: {arie,chechik}@cs.toronto.edu

Abstract. Multi-valued model-checking is an extension of classical model-checking to reasoning about systems with uncertain information, which are common during early design stages. The additional values of the logic are used to capture the degree of uncertainty. In this paper, we show that the multi-valued μ-calculus model-checking problem is reducible to several classical model-checking problems. The reduction allows one to reuse existing model-checking tools and algorithms to solve multi-valued model-checking problems. This paper generalizes, extends and corrects previous work in this area, done in the context of 3-valued models, symbolic model-checking, and De Morgan algebras.

1 Introduction

Temporal logic model-checking [CGP99] is one of the most widely used automated verification techniques. Its strength lies in its "push-button" approach to reasoning. Once a user has specified a model K, usually as a finite-state transition system, and a property in some temporal logic L, a model-checker returns true if the model satisfies the property and false otherwise.

In this paper, we assume that the temporal logic used to specify properties is the temporal μ-calculus [Koz83]. Classical model-checking is defined over concrete models that explicitly allow some behaviors and prohibit others. This makes it well suited for analyzing systems at the end of the design cycle when all of the information is known. However, it is inconvenient for models that contain uncertain information, which is common during early design stages.

Sources of uncertainty come from partial information about the system, or internal inconsistencies. The former include partial systems where some behaviors are neither explicitly allowed nor prohibited, and abstracted systems where an abstraction results in the loss of information. Another source of uncertain information can be the property itself. For example, in temporal logic query-checking [Cha00], temporal logic is extended with unknowns (called *placeholders*) that indicate a user's uncertainty about the correct formulation of the property. Inconsistent models come from representing a system as a composition of several (usually consistent) modules. Such modules can be features, with the goal of discovering feature interaction, or partial descriptions of the system, contributed by different stakeholders. In both cases, inconsistencies are inevitable.

Multi-valued logics provide a unifying framework for reasoning about systems with uncertain information [Fit92,Fit91,CDEG03]. Additional logic values are used to capture the degree of uncertainty, and are used to construct the model of the system. For example, partial information can be represented using a 3-valued logic [Kle52] with values T, M and F, where T and F represent definite information, and M ("maybe") represents partial knowledge [BG99].

Multi-valued logics are typically defined using finite De Morgan algebras [Ras78], also known as quasi-boolean algebras. This ensures that many laws of classical logic, such as idempotance, associativity, distributivity, De Morgan laws, involution of negation ($\neg\neg a = a$), are preserved. Laws that are not necessarily preserved include non-contradiction ($a \wedge \neg a = \bot$) and excluded middle ($a \vee \neg a = \top$). *Multi-valued model-checking* [CDEG03] is defined as a procedure that receives a multi-valued transition system K (where either propositions are multi-valued, the transition relation is multi-valued, or both) and a formula φ from a temporal logic defined over some De Morgan algebra $\mathcal{L}$, and returns the degree to which φ holds on K.

The multi-valued model-checking problem can be decided directly, using a specialized tool [CDG02]. Yet, it is appealing to reduce it to several classical model-checking problems. Such a reduction allows one to check correctness of the direct approach and opens venues to use the mature classical model-checking technology. It also provides a connection between multi-valued and classical model-checking and allows to lift theoretical results from classical model-checking to multi-valued. For example, it is used in [HJS01,HJS03] to show that the refinement relation over 3-valued models is an extended version of the bisimulation relation [Mil89], and in [GCD03] to show that query-checking is an instance of multi-valued model-checking.

Reduction algorithms for 3-valued logic [BG00,BG99,GHJ01,HJS01,HP03] have been well understood. Such reductions typically involve two independent checks to classical models, and the negation is handled on the level of atomic propositions. Konikowska and Penczek [KP02a,KP02b] provided reductions for several other logics for the negation-free fragment of mv-$\mathrm{CTL}^*(\mathcal{L})$. The contribution of this paper is in generalizing these reductions to μ-calculus over arbitrary finite De Morgan algebras. The solution is effectively to reduce multi-valued model-checking to $|\mathcal{J}(\mathcal{L})|$ 2-valued models, where $\mathcal{J}(\mathcal{L})$ is the set of join-irreducible elements of a given De Morgan algebra $\mathcal{L}$. Each 2-valued model is encoded to be able to decide both universal and existential temporal logic properties. A similar approach, although in the context of *classical* temporal logic questions, was proposed by Huth and Pradhan [HP03].

The rest of the paper is organized as follows. After giving the necessary background in Section 2, we systematically develop and analyze the reduction algorithm in Section 3 and compare it with related work in Section 4. Section 5 concludes the paper.

2 Background

In this section, we give a brief introduction to lattice theory, De Morgan algebras, and multi-valued model-checking.

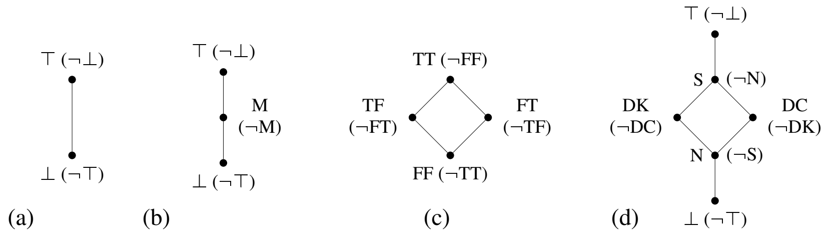

Fig. 1. Examples of a few distributed lattices and the corresponding De Morgan algebras (bracketed values describe the negation): (a) lattice $\mathbf{2}$; (b) lattice $\mathbf{3}$; (c) cross-product lattice $\mathbf{2x2}$; and (d) lattice $\mathbf{2x2} + \mathbf{2}$.

2.1 Lattices and De Morgan Algebras

A *lattice* is a partial order $(\mathcal{L}, \sqsubseteq)$, where every finite subset $B \subseteq \mathcal{L}$ has a least upper bound (called "join" and written $\sqcup B$) and a greatest lower bound (called "meet" and written $\sqcap B$). $\top$ and $\bot$ are the maximal and the minimal elements of a lattice, respectively. For notational convenience, we often refer to a lattice $(\mathcal{L}, \sqsubseteq)$ by its carrier set $\mathcal{L}$. A lattice is called *distributive* if meet and join distribute over each other, i.e., $a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$. A few examples of distributive lattices are given in Figure 1.

Definition 1. *[DP90] An element j in a lattice $\mathcal{L}$ is* join-irreducible *iff $j \neq \bot$ and for any x and y in $\mathcal{L}$, $j = x \sqcup y$ implies $j = x$ or $j = y$.*

In other words, j is a join-irreducible if it cannot be further decomposed into a join of other elements in the lattice. For example, the join-irreducibles of the lattices in Figure 1(a), (b), (c), (d) are $\{\top\}$, $\{\top, \text{M}\}$, $\{\text{TF,FT}\}$, $\{\top, \text{ DK, DC, N}\}$, respectively. The set of all join-irreducibles of $\mathcal{L}$ is denoted by $\mathcal{J}(\mathcal{L})$.

Every element of a finite lattice can be uniquely decomposed as a join of all join-irreducible elements below it:

Theorem 1. *[DP90] For any $\ell \in \mathcal{L}$, $\ell = \bigsqcup\{j \in \mathcal{J}(\mathcal{L}, \sqsubseteq) \mid j \sqsubseteq \ell\}$.*

For any join-irreducible element $j \in \mathcal{J}(\mathcal{L})$, the function $\cdot \sqsupseteq j$ distributes over meets and joins:

$$(a \sqcap b) \sqsupseteq j = (a \sqsupseteq j) \sqcap (b \sqsupseteq j) \qquad (a \sqcup b) \sqsupseteq j = (a \sqsupseteq j) \sqcup (b \sqsupseteq j)$$

For any lattice $\mathcal{L}$ and a collection of $\mathcal{L}$ elements B, the *downward closure* of B, written $\downarrow B$, is the set of all elements of $\mathcal{L}$ that are below some elements of B:

$$\downarrow B \triangleq \{\ell \in \mathcal{L} \mid \exists b \in B \cdot \ell \sqsubseteq b\}$$

Definition 2. *A De Morgan algebra is a tuple $(\mathcal{L}, \sqsubseteq, \neg)$, where $(\mathcal{L}, \sqsubseteq)$ is a finite distributive lattice and $\neg$ is any operation that preserves involution ($\neg\neg\ell = \ell$) and De Morgan laws.*

De Morgan algebras provide a natural model for De Morgan logics where the logical conjunction ($\wedge$) and disjunction ($\vee$) are interpreted as meet and join of the algebra, respectively. In De Morgan algebras, we get $\neg\top = \bot$ and $\neg\bot = \top$, but not necessarily the law of non-contradiction ($\ell \sqcap \neg\ell = \bot$) or excluded middle ($\ell \sqcup \neg\ell = \top$). For notational convenience, we write $\Rightarrow$ for material implication: $a \Rightarrow b \triangleq \neg a \sqcup b$.

We can define several De Morgan algebras using the lattices given in Figure 1. The domain of logical values of the classical logic, referred to as $\mathbf{2}$, is the lattice in Figure 1(a). The three-valued algebra $\mathbf{3}$ (Kleene logic [Kle52]) is defined on the lattice in Figure 1(b), where $\neg\top = \bot$, $\neg\bot = \top$, $\neg$M = M. The four-valued algebra $\mathbf{2x2}$ is defined on the lattice in Figure 1(c). A logic based on this algebra can be used for reasoning about inconsistency. Note that $\top$ and $\bot$ elements of an algebra are interpreted as values true and false of the logic, respectively. When the negation and the ordering operators of an algebra $(\mathcal{L}, \sqsubseteq, \neg)$ are clear from the context, we refer to it by its carrier set $\mathcal{L}$.

Given a set S, and a De Morgan algebra $\mathcal{L}$, we denote the set of functions from S to $\mathcal{L}$ by $\mathcal{L}^S$. If $\mathcal{L}$ is a De Morgan algebra, then so is $(\mathcal{L}^S, \sqsubseteq, \neg)$, where $\sqsubseteq$ and $\neg$ are pointwise extensions of the corresponding operators of $\mathcal{L}$. That is, for $\mathbb{F}, \mathbb{G} \in \mathcal{L}^S$,

$$\mathbb{F} \sqsubseteq \mathbb{G} = \forall s \in S \cdot \mathbb{F}(s) \sqsubseteq \mathbb{G}(s) \qquad \mathbb{G} = \neg\mathbb{F} \textbf{ iff } \forall s \in S \cdot \mathbb{G}(s) = \neg\mathbb{F}(s)$$

Theorem 2. *[CDE+01] For any De Morgan algebra $(\mathcal{L}, \sqsubseteq, \neg)$ there exists a function* $\mathsf{neg} : \mathcal{J}(\mathcal{L}) \to \mathcal{J}(\mathcal{L})$ *defined as* $\mathsf{neg}(j) \triangleq \sqcap(\mathcal{L} \setminus \downarrow \neg j)$*, such that*

$$\forall \ell \in \mathcal{L} \cdot \forall j \in \mathcal{J}(\mathcal{L}) \cdot \neg\ell \sqsupseteq j \;=\; \neg(\ell \sqsupseteq \mathsf{neg}(j))$$

Note that neg maps join-irreducible elements to join-irreducible elements and can be easily [CDE+01]. For example, for the algebra $\mathbf{3}$, $\mathsf{neg}(\top) = \mathrm{M}$ and $\mathsf{neg}(\mathrm{M}) = \top$. For the algebras $\mathbf{2}$ and $\mathbf{2x2}$, neg is the identity function.

2.2 Multi-Valued Model-Checking

Multi-valued model-checking [CEP01] is a generalization of the temporal logic model-checking problem to arbitrary De Morgan logics. A multi-valued model-checker receives a De Morgan algebra, a multi-valued model, and a temporal property, and determines the value with which this property holds in the model. Multi-valued models are defined over χKripke structures – generalizations of Kripke structures, where each atomic proposition and each transition between a pair of states are labeled with values from the algebra. Formally, a χKripke structure is a tuple $K = (S, s_0, \mathbb{R}, I, A, \mathcal{L})$, where S is a finite set of states; $\mathcal{L}$ is a De Morgan algebra; A is a set of atomic propositions; $s_0 \in S$ is the initial state; $\mathbb{R} : S \times S \to \mathcal{L}$ is a multi-valued transition relation; $I : S \times A \to \mathcal{L}$ is a (total) labeling function, such that for each atomic proposition $a \in A$, $I(s, a) = \ell$ means that variable a has value ℓ in state s. Thus, any Kripke structure is also a χKripke structure over the algebra $\mathbf{2}$. An example χKripke structure for the algebra $\mathbf{3}$ is given in Figure 2(a). To avoid clutter when presenting finite-state machines graphically, we follow the convention of not showing $\bot$ transitions and not labeling $\top$ transitions.

Temporal logic properties are specified in $L_\mu^+(A, \mathcal{L})$ – a generalization of μ-calculus [Koz83] to arbitrary De Morgan algebras.

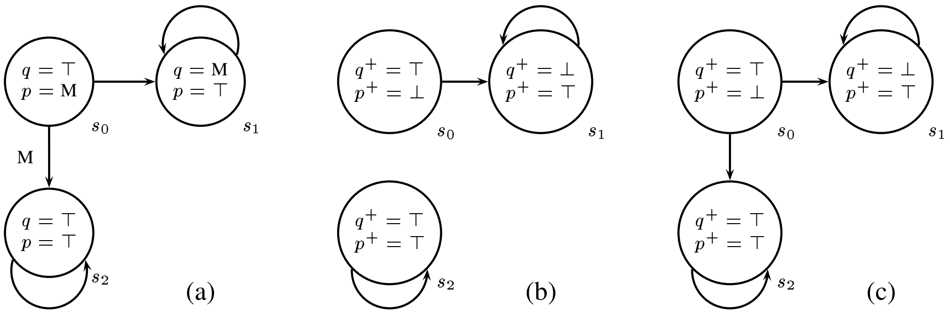

Fig. 2. (a) A χKripke structure K over the algebra $\mathbf{3}$; (b) a Kripke structure $K_\top$ used to check truth of existential properties over K; (c) a Kripke structure $K'_\top$ for universal properties.

Definition 3. *Let* Var *be a set of fixpoint variable names,* A *be a set of propositions, and* $\mathcal{L}$ *be a De Morgan algebra. The logic* $L^+_\mu(A, \mathcal{L})$ *is the set of formulas defined as:*

$$\varphi \triangleq Z \mid \ell \mid \neg\ell \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \nu Z \cdot \varphi \mid \mu Z \cdot \varphi$$

where $\ell \in \mathcal{L}$*,* $p \in A$*, and* $Z \in \mathsf{Var}$.

$\Diamond$ and $\Box$ are the next-state operators, with the intuitive meaning "there exists a next state" and "for all next states", respectively. This gives rise to $\Diamond L^+_\mu(A, \mathcal{L})$ (*existential*) and $\Box L^+_\mu(A, \mathcal{L})$ (*universal*) fragments in which the only allowed next-state operators are $\Diamond$ and $\Box$, respectively.

We write $\varphi(Z)$ for a $L^+_\mu(A, \mathcal{L})$ formula φ that *may* contain a free occurrence of Z, and $\varphi(\psi)$ for a formula obtained from φ by replacing *all* free occurrences of Z by ψ. μ and ν denote the least and the greatest fixpoint operators, respectively.

Note that in $L^+_\mu(A, \mathcal{L})$, the negation operator $\neg$ is restricted to elements of $\mathcal{L}$ and propositions. Alternatively, we can define a logic $L_\mu(A, \mathcal{L})$ by relaxing this restriction. In this case, for a formula $\varphi \in L_\mu(A, \mathcal{L})$, $\nu Z \cdot \varphi(Z)$ and $\mu Z \cdot \varphi(Z)$ are in $L_\mu(A, \mathcal{L})$ if and only if Z occurs under an even scope of negations in φ. For example, $\mu Z \cdot \neg\Diamond\neg Z$ is in $L_\mu(A, \mathcal{L})$, but $\mu Z \cdot \neg\Diamond Z$ is not. The traditional definition of μ-calculus is equivalent to $L_\mu(A, \mathbf{2})$. To simplify the notation, we often write L_μ when parameters A and $\mathcal{L}$ are clear from the context, or $L_\mu(A)$ and $L_\mu(\mathcal{L})$ when we want to emphasize only one of the parameters.

The semantics of L_μ is given by the function $||\cdot||$ that, for each formula φ and a state s of a χKripke structure, returns the value of φ in s. Note that $||\cdot||$ takes an additional parameter called an *environment* that is used to interpret the fixpoint variables.

Definition 4. *Let* ℓ *be an element of* $\mathcal{L}$*,* $p \in A$*,* $s \in S$*,* $\varphi, \psi \in L_\mu$*, and* $\rho : \mathsf{Var} \to \mathcal{L}^S$*. Then the function* $||\cdot|| : L_\mu \times (\mathsf{Var} \to \mathcal{L}^S) \to \mathcal{L}^S$ *is defined as follows:*

$$\begin{array}{rlrl}
||Z||_\rho(s) & \triangleq \rho(Z)(s) & ||\ell||_\rho(s) & \triangleq \ell \\
||p||_\rho(s) & \triangleq I(s, p) & ||\neg\varphi||_\rho(s) & \triangleq \neg||\varphi||_\rho(s) \\
||\varphi \wedge \psi||_\rho(s) & \triangleq ||\varphi||_\rho(s) \sqcap ||\psi||_\rho(s) & ||\varphi \vee \psi||_\rho(s) & \triangleq ||\varphi||_\rho(s) \sqcup ||\psi||_\rho(s) \\
||\Diamond\varphi||_\rho(s) & \triangleq \sqcup_{t \in S}(\mathbb{R}(s,t) \sqcap ||\varphi||_\rho(t)) & ||\Box\varphi||_\rho(s) & \triangleq \sqcap_{t \in S}(\mathbb{R}(s,t) \Rightarrow ||\varphi||_\rho(t))
\end{array}$$

$$||\mu Z \cdot \varphi(Z)||_\rho(s) \triangleq \sqcap\{\mathbb{C} \in \mathcal{L}^S \mid ||\varphi||_{\rho[Z \to \mathbb{C}]} \sqsubseteq \mathbb{C}\}(s)$$
$$||\nu Z \cdot \varphi(Z)||_\rho(s) \triangleq \sqcup\{\mathbb{C} \in \mathcal{L}^S \mid \mathbb{C} \sqsubseteq ||\varphi||_{\rho[Z \to \mathbb{C}]}\}(s)$$

where $\rho[Z \to \mathbb{C}]$ *is an environment like* ρ *except that it maps* Z *to* $\mathbb{C}$.

An environment that maps every $Z \in \mathsf{Var}$ to $\bot$ is denoted by $\bot$. For a closed L_μ formula φ, we write $||\varphi||$ to stand for $||\varphi||_\bot$. The value of a closed L_μ formula φ on a χKripke structure K is given by the value of φ in the initial state of K, i.e., $||\varphi||(s_0)$, and is often written as $||\varphi||^K$.

Note that under our definition, the next-time operators $\Diamond$ and $\Box$ are duals of each other, i.e., $\neg\Diamond\neg\varphi = \Box\varphi$. This also ensures the duality of the least and the greatest fixpoint operators, i.e. $\neg\mu Z \cdot \varphi(\neg Z) = \nu Z \cdot \varphi(Z)$. Combining the above results with the involution property of the negation operator, we obtain the following theorem.

Theorem 3. *The negation-free fragment L_μ^+ of L_μ is as expressive as L_μ.*

Both CTL [CES86] and its multi-valued extension χCTL$(\mathcal{L})$ [CDEG03] can be expressed in $L_\mu(A, \mathcal{L})$ as follows:

$$\begin{aligned} EXp &= \Diamond p & AXp &= \Box p \\ E[\varphi U \psi] &= \mu Z \cdot \psi \vee \varphi \wedge \Diamond Z & A[\varphi U \psi] &= \mu Z \cdot \psi \vee \varphi \wedge \Box Z \\ EG\varphi &= \nu Z \cdot \varphi \wedge \Diamond Z & AG\varphi &= \nu Z \cdot \varphi \wedge \Box Z \end{aligned}$$

So, the reduction technique developed later in this paper for $L_\mu(\mathcal{L})$ is directly applicable to χCTL$(\mathcal{L})$ as well.

3 Reduction

In this section, we systematically decompose a multi-valued μ-calculus (L_μ) model-checking problem into several classical μ-calculus ($L_\mu(\mathbf{2})$) model-checking problems. One approach to the reduction, particularly prevalent for model-checking over the algebra **3** (e.g. [BG99]), is to reduce the model only, while keeping the formula the same. In the 3-valued case, one constructs two models, corresponding to "the best" and "the worst" possible behaviors, with respect to the given temporal property. Of course, the definition of "best" and "worst" depends on quantifiers used in the property; further, for a property containing both universal and existential quantifiers, one must reduce the formula as well!

We start by showing that a model-checking problem for $L_\mu(A, \mathcal{L})$ is reducible to several model-checking problems over a different logic, which we call $L_{\overline{\mu}}^{\sqsupseteq}(A, \mathcal{L})$ (Section 3.1). We then show how to change a χKripke structure so that each of the resulting problems can be solved using a single call to a classical model-checker on a Kripke structure (Section 3.2). We put the two reductions together and illustrate them on an example in Section 3.3. Section 3.4 summarizes consequences of the reduction and analyzes its complexity.

3.1 Model-Checking: From Multi-Valued to Boolean

This section introduces the first step of the reduction, by showing that a multi-valued model-checking problem can be reduced to several boolean model-checking problems. Note that this step only changes the property, while leaving the model unchanged.

For any De Morgan algebra $\mathcal{L}$, any element $\ell \in \mathcal{L}$ is uniquely represented by the join of join-irreducible elements below it (Theorem 1). For any $L_\mu(\mathcal{L})$ formula φ and state s, $||\varphi||(s)$ is simply an element of $\mathcal{L}$, so we can extend Theorem 1 to $L_\mu(\mathcal{L})$ as follows:

Theorem 4. *Let φ be a $L_\mu(\mathcal{L})$ formula, and s be a state of a χKripke structure over $\mathcal{L}$. Then $||\varphi||_\rho(s) = \bigsqcup_{j \in \mathcal{J}(\mathcal{L})}(j \sqcap (||\varphi||_\rho(s) \sqsupseteq j))$.*

This theorem provides the basis for our reduction technique. The expression $||\varphi||_\rho(s) \sqsupseteq j$ is interpreted over a χKripke structure, but it always evaluates to either $\top$ or $\bot$. Thus, it allows us to reduce a multi-valued model-checking problem for $L_\mu(\mathcal{L})$ to $|\mathcal{J}(\mathcal{L})|$ boolean model-checking problems.

In order to express the statement $||\varphi||_\rho(s) \sqsupseteq j$ by a single temporal logic formula, we introduce the logic $L_\mu^{\sqsupseteq}$.

Definition 5. *Let Var be a set of fixpoint variable names, A be a set of propositions, and $\mathcal{L}$ be a De Morgan algebra. The logic $L_\mu^{\sqsupseteq}(A, \mathcal{L})$ is the set of formulas defined as:*

$$\varphi \triangleq Z \sqsupseteq j \mid \top \mid \bot \mid p \sqsupseteq j \mid \neg p \sqsupseteq j \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\sqsupseteq j]\varphi \mid \langle \sqsupseteq j \rangle \varphi \mid \nu Z \cdot \varphi \mid \mu Z \cdot \varphi$$

where $j \in \mathcal{L}$, $p \in A$, and $Z \in \mathsf{Var}$.

Furthermore, in expressions $Z \sqsupseteq \ell_1, \dots Z \sqsupseteq \ell_n$, we require that all algebra values be the same, i.e. $\forall i, j \cdot \ell_i = \ell_j$. The semantics of $L_\mu^{\sqsupseteq}(A, \mathcal{L})$ is given with respect to χKripke structures and is defined as follows:

Definition 6. *Let j be elements of $\mathcal{L}$, $p \in A$, $s \in S$, $\varphi, \psi \in L_\mu^{\sqsupseteq}$, and $\rho : \mathsf{Var} \to \mathcal{L}^S$. Then the function $|| \cdot || : L_\mu^{\sqsupseteq} \times (\mathsf{Var} \to \mathcal{L}^S) \to \mathcal{L}^S$ is defined as:*

$$||\top||_\rho(s) \triangleq \top \qquad\qquad ||\bot||_\rho(s) \triangleq \bot$$
$$||p \sqsupseteq j||_\rho(s) \triangleq I(s,p) \sqsupseteq j \qquad\qquad ||\neg p \sqsupseteq j||_\rho(s) \triangleq \neg I(s,p) \sqsupseteq j$$
$$||Z \sqsupseteq j||_\rho(s) \triangleq \rho(Z)(s) \sqsupseteq j$$
$$||\langle \sqsupseteq j \rangle \varphi||_\rho(s) \triangleq \sqcup_{t \in S}((\mathbb{R}(s,t) \sqsupseteq j) \sqcap ||\varphi||_\rho(t))$$
$$||[\sqsupseteq j]\varphi||_\rho(s) \triangleq \sqcap_{t \in S}((\mathbb{R}(s,t) \sqsupseteq j) \Rightarrow ||\varphi||_\rho(t))$$

with the semantics of $\wedge$, $\vee$, μ and ν operators being the same as in Definition 4.

Finally, we show that for any $L_\mu(\mathcal{L})$ formula φ, and any join-irreducible element j, the statement $||\varphi||_\rho \sqsupseteq j$ is expressible in $L_\mu^{\sqsupseteq}$.

Theorem 5. *Let φ be a $L_\mu(\mathcal{L})$ formula, $j \in \mathcal{J}(\mathcal{L})$, and s be a state of a χKripke structure. Then there exists a $L_\mu^{\sqsupseteq}$ formula $\varphi \Uparrow j$, called the* cut *of φ with respect to j, such that $||\varphi||_\rho(s) \sqsupseteq j \; = \; ||\varphi \Uparrow j||_\rho(s)$.*

The proof of this theorem is given in the Appendix. As a direct consequence of the proof, we obtain the following procedure for constructing $\varphi \Uparrow j$. Given a formula $\varphi \in L_\mu(\mathcal{L})$ and a join-irreducible element j of $\mathcal{L}$, $\varphi \Uparrow j$ is constructed by recursively applying the transformation $\cdot \Uparrow j$ that distributes over $\wedge$, $\vee$, and greatest and least fixpoints μ and ν:

$$p \Uparrow j = p \sqsupseteq j \qquad\qquad (\neg p) \Uparrow j = \neg p \sqsupseteq j$$
$$(\Diamond \varphi) \Uparrow j = \langle \sqsupseteq j \rangle (\varphi \Uparrow j) \qquad\qquad (\Box \varphi) \Uparrow j = [\sqsupseteq \mathsf{neg}(j)](\varphi \Uparrow j)$$

For example, if $\varphi = \mu Z \cdot p \vee (\Box Z \wedge \Diamond \top)$, then its cut with respect to a join-irreducible element $\top$ of the algebra **3** is given by

$$\varphi \Uparrow \top = \mu Z \cdot (p \sqsupseteq \top) \vee ([\sqsupseteq \mathrm{M}](Z \sqsupseteq \top) \wedge \langle \sqsupseteq \top \rangle (\top \sqsupseteq \top))$$

Combining Theorem 4 and Theorem 5, we obtain the following theorem.

Theorem 6. *A multi-valued model-checking problem for the logic $L_\mu(\mathcal{L})$ is reducible to $\mathcal{J}(|\mathcal{L}|)$ boolean model-checking problems for $L_\mu^{\sqsupseteq}(\mathcal{L})$.*

3.2 Reducing $L_\mu^{\sqsupseteq}$

We now show that the problem of model-checking a formula $\varphi \in L_\mu^{\sqsupseteq}$ on a χKripke structure is reducible to a *single* model-checking problem for classical μ-calculus $L_\mu(\mathbf{2})$ on a Kripke structure.

Theorem 7. *Let φ be a $L_\mu^{\sqsupseteq}(A, \mathcal{L})$ formula, $K = (S, s_0, \mathbb{R}, I, A, \mathcal{L})$ be a χKripke structure, and ρ be an environment. Then there exists a $L_\mu(A', \mathbf{2})$ formula $T(\varphi)$, a Kripke structure $K' = (S', s_0', \mathbb{R}', I', A', \mathbf{2})$, and an environment ρ' such that $||\varphi||_\rho^K = ||T(\varphi)||_{\rho'}^{K'}$. Moreover, $|S'|$ is in $O(|\mathcal{J}(\mathcal{L})| \times |S|)$.*

Thus, given an algebra $\mathcal{L}$, the model-checking problem for $L_\mu^{\sqsupseteq}(\mathcal{L})$ is reducible to model-checking a $L_\mu(\mathbf{2})$ formula at the expense of a linear increase in the size of the statespace. The Kripke structure K' is obtained from K by first constructing a Kripke Transition system (KTS) [MOSS99], treating algebra values on transitions as actions, and then converting the resulting KTS into a Kripke structure.

Instead of proving Theorem 7 in the general case, we prove it for two fragments of $L_\mu^{\sqsupseteq}$ that are used in the reduction in Section 3.1.

Temporal logic $L_\mu^{\sqsupseteq j}$. Let j be an element of $\mathcal{L}$. Then the fragment $L_\mu^{\sqsupseteq j}$ is defined to be the set of all formulas of $L_\mu^{\sqsupseteq}$ where only j can appear on the right-hand side of $\sqsupseteq$, and only $\langle \sqsupseteq j \rangle$ and $[\sqsupseteq j]$ are allowed. For example, $\nu Z \cdot (p \sqsupseteq \mathrm{M}) \vee \langle \sqsupseteq \mathrm{M} \rangle (Z \sqsupseteq \mathrm{M})$ is in $L_\mu^{\sqsupseteq \mathrm{M}}(\{p\}, \mathbf{3})$, but $(p \sqsupseteq \top)$ is not. This fragment is used to reduce existential properties.

Given a formula $\varphi \in L_\mu^{\sqsupseteq j}(A, \mathcal{L})$ and a χKripke structure $K = (S, s_0, \mathbb{R}, I, A, \mathcal{L})$, we construct a Kripke structure $K' = (S, s_0, \mathbb{R}', I', A', \mathbf{2})$ as follows:

$$
\begin{aligned}
A' &\triangleq \{p^+ \mid p \in A\} \cup \{p^- \mid p \in A\} & \qquad I'(s, p^+) &\triangleq I(s, p) \sqsupseteq j \\
\mathbb{R}'(s, t) &\triangleq \mathbb{R}(s, t) \sqsupseteq j & I'(s, p^-) &\triangleq \neg I(s, p) \sqsupseteq j
\end{aligned}
$$

Note that K' has the same statespace as K, but twice as many propositions. For every proposition $p \in A$, it has a pair of propositions p^+ and p^- corresponding to p and $\neg p$, respectively. The transition relation of K' consists of all transitions of K whose value is above j. The reduced formula $T(\varphi)$ is obtained from φ by recursively removing $\sqsupseteq j$ and replacing every occurrence of $\langle \sqsupseteq j \rangle$ and $[\sqsupseteq j]$ by $\Diamond$ and $\Box$, respectively. For example, a formula $\varphi = \nu Z \cdot (p \sqsupseteq j) \wedge [\sqsupseteq j](Z \sqsupseteq j)$ is reduced to $T(\varphi) = \nu Z \cdot p^+ \wedge \Box Z$. Finally, an environment ρ is replaced by $\rho' \triangleq \rho \sqsupseteq j$. The fact that $||\varphi||_\rho^K = ||T(\varphi)||_{\rho'}^{K'}$ follows trivially from the construction.

Temporal logic $L_\mu^{j,i}$. Let j and i be elements of $\mathcal{L}$. Then the fragment $L_\mu^{j,i}$ is defined to be the set of all formulas obtained from $L_\mu^{\sqsupseteq j}$ by replacing the universal next-time operator $[\sqsupseteq j]$ with $[\sqsupseteq i]$. For example, $[\sqsupseteq M](p \sqsupseteq \top)$ is in $L_\mu^{\top, M}(\{p\}, \mathbf{3})$, but $[\sqsupseteq \top](p \sqsupseteq \top)$ is not. This fragment is used to reduce properties that contain both universal and existential next-time operators.

Given a formula $\varphi \in L_\mu^{j,i}(A, \mathcal{L})$ and a χKripke structure $K = (S, s_0, \mathbb{R}, I, A, \mathcal{L})$, we construct a Kripke structure $K' = (S \times \{\top, \bot\}, (s_0, \top), \mathbb{R}', I', A', \mathbf{2})$ as follows:

$$
\begin{array}{ll}
I'((s,a), p^+) \triangleq I(s,p) \sqsupseteq j & \mathbb{R}'((s,a),(t,\top)) \triangleq \mathbb{R}(s,t) \sqsupseteq j \\
I'((s,a), p^-) \triangleq \neg I(s,p) \sqsupseteq j & \mathbb{R}'((s,a),(t,\bot)) \triangleq \mathbb{R}(s,t) \sqsupseteq i \\
I'((s,a), tval^j) \triangleq a & I'((s,a), tval^i) \triangleq \neg a \\
\multicolumn{2}{c}{A' \triangleq \{p^+ \mid p \in A\} \cup \{p^- \mid p \in A\} \cup \{tval^j, tval^i\}}
\end{array}
$$

The Kripke structure K' can be seen as having two distinct transition relations $\mathbb{R} \sqsupseteq j$ and $\mathbb{R} \sqsupseteq i$. To encode this, its statespace is extended to *double* the size of the statespace of K such that there exists a transition between (s, a) and $(t, \top)$ if and only if there is a transition between s and t in K with value above j; and there exists a transition between (s, a) and $(t, \bot)$ if and only if there exists a transition between s and t in K with value above i. Similarly, its set of propositions A' is extended with additional propositions $tval^j$ and $tval^i$, where $tval^j$ is true in a state (s, a) if and only if s is reachable by a transition in K with value above j, and $tval^i$ is true if and only if s is reachable by a transition whose value is above i.

The reduced formula $T(\varphi)$ is obtained by eliminating all occurrences of $\sqsupseteq j$ from φ and replacing the temporal next-time operators $\langle \sqsupseteq j \rangle$ and $[\sqsupseteq i]$ as follows:

$$T(\langle \sqsupseteq j \rangle \varphi) = \Diamond(tval^j \wedge T(\varphi)) \qquad T([\sqsupseteq i]\varphi) = \Box(tval^i \Rightarrow T(\varphi))$$

Finally, an environment ρ is replaced by $\rho' = \rho \sqsupseteq j$. The proof that $||\varphi||_\rho^K = ||T(\varphi)||_{\rho'}^{K'}$ follows trivially from the construction.

Note that the logic $L_\mu^{j,i}$ is as expressive as $L_\mu^{\sqsupseteq j}$ for χKripke structures whose transition relation is boolean, i.e., $\forall s, t \in S \cdot \mathbb{R}(s,t) \in \{\top, \bot\}$. In this case, the operators $[\sqsupseteq j]$ and $[\sqsupseteq i]$ are identical for all i and j.

3.3 Model-Checking: From Multi-Valued to Classical

Here, we combine Theorems 6 and 7 to yield the overall reduction and illustrate it on an example.

Theorem 8. *A multi-valued model-checking problem for the logic $L_\mu(\mathcal{L})$ is reducible to $\mathcal{J}(|\mathcal{L}|)$ classical model-checking problems for $L_\mu(\mathbf{2})$.*

Theorem 8 leads to the following reduction algorithm. Given an $L_\mu(\mathcal{L})$ formula φ and a χKripke structure K, for every join-irreducible j we (a) construct the j-cut $\varphi \Uparrow j$, and (b) use one of the reductions of Section 3.2 to reduce checking $||\varphi \Uparrow j||^K$ to a classical model-checking problem. The choice of the reduction to use depends on the structure of φ: if φ is existential, its cut is expressible in $L_\mu^{\sqsupseteq j}$; otherwise, it is expressible in $L_\mu^{j,i}$.

Example. To illustrate the reduction algorithm, we apply it to an existential property $\varphi = \mu Z \cdot p \vee \Diamond Z$ and state s_0 of the χKripke structure K shown in Figure 2(a). We start by constructing cuts of φ with respect to the two join-irreducible elements of algebra $\mathbf{3}$:

$$
\begin{aligned}
\varphi \Uparrow \top &= \mu Z \cdot p \sqsupseteq \top \vee \langle \sqsupseteq \top \rangle (Z \sqsupseteq \top) \\
\varphi \Uparrow \mathrm{M} &= \mu Z \cdot p \sqsupseteq \mathrm{M} \vee \langle \sqsupseteq \mathrm{M} \rangle (Z \sqsupseteq \mathrm{M})
\end{aligned}
$$

Next, for each cut $\varphi \Uparrow j$, we construct a Kripke structure K_j and a μ-calculus formula corresponding to the cut. Following the construction outlined in Section 3.2, both cuts are reduced to $\mu Z \cdot p^+ \vee \Diamond Z$. The Kripke structure $K_\top$ shown in Figure 2(b) is obtained from K by eliminating all non-$\top$ transitions, and replacing atomic propositions by their positive and negative versions. For conciseness we only show positive propositions in the figure. Finally, we model-check the μ-calculus formula in state s_0 of $K_\top$. In our example, the property $\mu Z \cdot p^+ \vee \Diamond Z$ is true on $K_\top$ which implies that $\varphi \Uparrow \top$ holds on the χKripke structure K, and therefore the value of the original property φ on K is $\top$:

$$||\mu Z \cdot p^+ \vee \Diamond Z||^{K_\top} = \mathsf{true} \;\Leftrightarrow\; ||\varphi \Uparrow \top||^K = \top \;\Leftrightarrow\; ||\varphi||^K \sqsupseteq \top \;\Leftrightarrow\; ||\varphi||^K = \top$$

The case of $\varphi \Uparrow \mathrm{M}$ is similar, except that K_M is constructed from K by treating all non-$\perp$ transitions as $\top$.

For another example, let ψ be a universal property $\psi = \nu Z \cdot p \;\wedge\; \Box Z$. After computing the cuts and performing the reduction, we obtain $\nu Z \cdot p^+ \;\wedge\; \Box Z$. The Kripke structure $K'_\top$ corresponding to the $\top$-cut (see Figure 2(c)) is obtained from K by treating all non-$\perp$ transitions as $\top$. Model-checking the property on $K'_\top$ yields false, which implies that the cut $\psi \Uparrow \top$ evaluates to $\perp$ on the χKripke structure K, and therefore the value of ψ on K is less then $\top$. Since the algebra **3** has only three elements, this means that ψ evaluates to either M or $\perp$ on it.

$$||\nu Z \cdot p^+ \wedge \Box Z||^{K'_\top} = \mathsf{false} \;\Leftrightarrow\; ||\psi \Uparrow \top||^K = \perp \;\Leftrightarrow\; ||\psi||^K \not\sqsupseteq \top \;\Leftrightarrow\; ||\psi||^K \in \{\perp, \mathrm{M}\}$$

In this particular example, the value of ψ on K is M, and is obtained by checking the second cut, $\psi \Uparrow \mathrm{M}$. ☐

Note that in the example above, the cut properties $\varphi \Uparrow j$ for both join-irreducible elements were syntactically equivalent. Thus, we only had to reduce the χKripke structure, once for each join-irreducible element. However, the Kripke structures $K_\top$ and $K'_\top$, corresponding to the join-irreducible $\top$, were different: although they had the same statespace and the labeling function, the transition relation of $K_\top$ was that of K'_M. The reason is that $K_\top$ and $K'_\top$ were used to decide existential and universal formulas, respectively. In general, an existential part of a mixed formula should be checked over $K_\top$, and its universal part – over $K'_\top$. Then the Kripke structure corresponding to the join-irreducible $\top$ contains transition relations of *both* $K_\top$ and $K'_\top$, as in the second reduction in Section 3.2. This construction gets reflected in cut formulas, which are different for each join-irreducible.

3.4 Discussion and Complexity

We summarize the consequence of Theorem 8 for several fragments of $L_\mu(\mathcal{L})$ in Table 1. The first column of the table indicates the fragment of $L_\mu(\mathcal{L})$ used to specify the property; the second describes the restrictions placed on χKripke structures; the third specifies the number of $L_\mu(\mathbf{2})$ model-checking problems required; and the last indicates the ratio between the size of the statespace S' of the Kripke structures used by the reduction, and the statespace S of the original χKripke structure. For example, model-checking an arbitrary $L_\mu(\mathcal{L})$ property on a χKripke structure K is reducible to $|\mathcal{J}(\mathcal{L})|$ classical model-checking problems, each over a Kripke structure whose statespace is

Property	χKripke restrictions	# of $L_\mu(\mathbf{2})$ problems	$\frac{\|S'\|}{\|S\|}$
$L_\mu(\mathcal{L})$	none	$\|\mathcal{J}(\mathcal{L})\|$	2
	$\{\top, \bot\}$ transition relation		1
$\Diamond L_\mu(\mathcal{L})$	none		1
$\Box L_\mu(\mathcal{L})$	none		1

Table 1. Reducing multi-valued model-checking to classical.

twice that of K. On the other hand, if the property is expressed in either existential or universal fragments of $L_\mu(\mathcal{L})$, then the multi-valued model-checking problem is reducible to $|\mathcal{J}(\mathcal{L})|$ classical model-checking problems, each over a Kripke structure with the statespace identical to the statespace of the original χKripke structure.

Note that we have only considered reductions for the negation-free fragment $L_\mu^+(\mathcal{L})$ of $L_\mu(\mathcal{L})$. This is not a limitation of our approach since the negation-free fragment is as expressive as $L_\mu(\mathcal{L})$ (Theorem 3). Alternatively, it is easy to show that $||\neg\varphi \Uparrow j||(s) = \neg||\varphi \Uparrow \mathsf{neg}(j)||(s)$ directly:

$$\begin{array}{rlr} & ||\neg\varphi \Uparrow j||(s) & \text{(Definition of } \cdot \Uparrow j) \\ = & ||\neg\varphi||(s) \sqsupseteq j & \text{(Definition of } ||\cdot||) \\ = & \neg||\varphi||(s) \sqsupseteq j & \text{(Theorem 2)} \\ = & \neg(||\varphi||(s) \sqsupseteq \mathsf{neg}(j)) & \text{(Definition of } \cdot \Uparrow j) \\ = & \neg(||\varphi \Uparrow \mathsf{neg}(j)||(s)) & \end{array}$$

This, however, does not yield an elegant reduction algorithm.

4 Related Work

In this section, we compare the reduction presented in Section 3 to the work of others.

Multi-Valued Models (Fitting). Fitting [Fit91,Fit92] introduced a concept of multi-valued models and extended the propositional modal logic (i.e. $L_\mu(\mathcal{L})$ without the fixpoint operators) to them. In his models, the values of propositions and transitions come from a Heyting instead of a De Morgan algebra.

Definition 7. *[Fit92] A* Heyting algebra *is a tuple* $(\mathcal{L}, \sqsubseteq, \rightarrow, -)$*, where* $(\mathcal{L}, \sqsubseteq)$ *is a distributive lattice;* $\rightarrow$ *is a* relative pseudo-complement *defined as* $a \rightarrow b \triangleq \bigsqcup\{c \mid (c \sqcap a) \sqsubseteq b\}$*; and "$-$" is the negation operator defined as* $-a \triangleq a \rightarrow \bot$.

Heyting algebras are traditionally used as models for intuitionistic logic [Fit69], with the relative pseudo-complement operator $\rightarrow$ used to model intuitionistic implication. The negation operator "$-$" satisfies the law of non-contradiction ($a \wedge -a = \bot$), but not necessarily the law of excluded middle, the involution of negation, or any of the De Morgan laws.

Since the definition of $\Diamond L_\mu(\mathcal{L})$ only depends on the fact that $\mathcal{L}$ is distributive, it extends the propositional modal logic of [Fit92] with the fixpoint operators. For this logic, the reduction from multi-valued to two-valued semantics suggested by Fitting is identical to ours, with the only exception that he uses an equivalent concept of *proper prime filters* [DP90] instead of join-irreducible elements.

The definition of the universal next-time operator $\Box$ in [Fit92] is the same as ours syntactically ($||\Box\varphi||(s) \triangleq \sqcap_{t\in S}(\mathbb{R}(s,t) \Rightarrow ||\varphi||(t))$). However, Fitting interprets the implication $\Rightarrow$ operator as the relative pseudo-complement, whereas we interpret it as material implication. The two definitions coincide for Boolean algebras – algebras that are both Heyting and De Morgan.

Definition 8. *A* Boolean algebra *is a Heyting algebra* $(\mathcal{L}, \sqsubseteq, \rightarrow, -)$ *such that* $(\mathcal{L}, \sqsubseteq, -)$ *is a De Morgan algebra.*

The relative pseudo-complement $\rightarrow$ operator of a Boolean algebra is equivalent to the material implication $a \rightarrow b \;=\; a \Rightarrow b \;=\; -a \vee b$.

Note that in the special case of Boolean algebras, Theorem 2 can be strengthened as follows:

Theorem 9. *Let* $(\mathcal{L}, \sqsubseteq, \rightarrow, \neg)$ *be a Boolean algebra. Then for any join-irreducible element* $j \in \mathcal{J}(\mathcal{L})$*, and* $\ell \in \mathcal{L}$*:* $\neg\ell \sqsupseteq j \;=\; \neg(\ell \sqsupseteq j)$

A stronger version of Theorem 8 for Boolean algebras is given below.

Theorem 10. *For a Boolean algebra* $\mathcal{L}$*, multi-valued model-checking for the temporal logic* $L_\mu(\mathcal{L})$ *is reducible to* $|\mathcal{J}(\mathcal{L})|$ *model-checking problems for* $L_\mu^{\sqsupseteq j}$*.*

That is, for a Boolean algebra $\mathcal{L}$, we get $|\mathcal{J}(\mathcal{L})|$ classical model-checking problems, each over a Kripke structure with the statespace *identical* to the statespace of K.

Reducing Multi-Valued Model-Checking to Classical Model-Checking. Konikowska and Penczek [KP02a,KP02b] introduced a multi-valued temporal logic mv-$\mathrm{CTL}^*(\mathcal{L})$ – an extension of CTL^* [Eme90] to De Morgan algebras – and defined its semantics on χKripke structures. An interesting consequence of their definition is that it does not preserve the duality of a universal path quantifier A and an existential path quantifier E. For example, for an mv-$\mathrm{CTL}^*(\mathcal{L})$ state formula φ

$$\begin{array}{ll} \quad ||A\varphi||(s) & \text{(Definition of } ||\cdot|| \text{ from [KP02b])} \\ = \sqcap_{t\in S}(\mathbb{R}(s,t) \sqcap ||\varphi||(t) & \\ \neq \neg \bigsqcup_{t\in S}(\mathbb{R}(s,t) \sqcap ||\neg\varphi||(t) & \text{(Definition of } ||\cdot|| \text{ from [KP02b])} \\ = ||\neg E\neg\varphi||(s) & \end{array}$$

The consequences of this observation are: (a) when restricted to the classical logic, mv-$\mathrm{CTL}^*(\mathcal{L})$ is equivalent to the classical CTL^* *only* on *total* Kripke structures; (b) contrary to the claim made in [KP02b], the negation-free fragment of mv-$\mathrm{CTL}^*(\mathcal{L})$ is *less* expressive than the full mv-$\mathrm{CTL}^*(\mathcal{L})$; and (c) mv-$\mathrm{CTL}^*(\mathcal{L})$ does not subsume commonly used multi-valued temporal logics such as χCTL($\mathcal{L}$) and the 3-valued μ-calculus of [BG99].

In [KP02b], Konikowska and Penczek also identified sufficient conditions for reducing a model-checking problem for the negation-free fragment of mv-$\mathrm{CTL}^*(\mathcal{L})$ into several mv-CTL^* model-checking problems over sub-algebras of $\mathcal{L}$. However, neither a proof of the existence of the reduction, nor a constructive algorithm for it are provided. Instead, the reduction is developed for three classes of De Morgan algebras: a finite total order, a product of finite total orders, and a De Morgan algebra depicted in Figure 1(d). Theorem 8 in the current paper provides the missing existence proof and the reduction algorithm.

Chechik et al. [CDE+01] developed a symbolic model-checking algorithm for χCTL($\mathcal{L}$) based on Binary Decision Diagrams (BDD) [Bry86]. They showed that for a given set S and a De Morgan algebra $\mathcal{L}$, $S \rightarrow \mathcal{L}$ functions (an mv-set in the terminology of [CDE+01]) can be represented and manipulated using a collection of $|\mathcal{J}(\mathcal{L})|$ boolean functions (or sets). For a given mv-set $\mathbb{Z} : S \rightarrow \mathcal{L}$, a set $\mathbb{Z} \Uparrow j$ of the collection corresponding to a join-irreducible element $j \in \mathcal{J}(\mathcal{L})$ is defined as $\mathbb{Z} \Uparrow j \triangleq \{x \in S \mid \mathbb{Z}(x) \sqsupseteq j\}$. In the current paper, we have extended the reduction technique of [CDE+01] to a richer logic $L_\mu(\mathcal{L})$ and decoupled the reduction from any particular implementation of the model-checking algorithm. Thus, our reduction technique is applicable to any current or future algorithm for classical model-checking together with any optimizations. The particular implementation of Chechik et al. [CDE+01] can be seen as an application of the reduction technique presented in the current paper in the context of symbolic model-checking.

Reductions for 3-valued reasoning. Bruns and Godefroid [BG00,BG99], Godefroid et. al [GHJ01], and Huth et al. [HJS01,HJS03] studied the problem of model-checking over the algebra **3** on a variety of 3-valued finite-state transition systems. Bruns and Godefroid [BG99,BG00] investigated 3-valued model-checking on Partial Kripke structures, where propositions are 3-valued but the transition relation is in $\{\top, \bot\}$. Godefroid et al. [GHJ01] provided an extension of the algorithm to Modal Transition Systems – a generalization of Labeled Transition Systems of [LT88], in which a transition relation is allowed to become 3-valued. The idea is further extended by Huth et al. [HJS01] to Kripke Modal Transition Systems which are equivalent to our χKripke structures when the algebra is **3**. In all of these cases, it is shown that 3-valued model-checking is reducible to two classical model-checking problems. This is not surprising since all of the modeling formalisms have been shown to be equivalent [GJ03]. Our logic $L_\mu(\mathbf{3})$ is equivalent to the 3-valued μ-calculus of [BG00,GHJ01,HJS01] (and thus subsumes 3-valued CTL [BG99]). So the multi-valued model-checking reduction technique presented here can be seen as an extension of the reduction for the 3-valued model-checking beyond the algebra **3**.

Partial Model Checking from Multiple Viewpoints. The work of Huth and Pradhan [HP03] is the closest to ours. In this work, each of C stakeholders, arranged in a partial order, submits a partial model, consisting of valid (“must”) and consistent (“may”) statements about states and transitions. Given a first-order property, the model-checking problem is to determine sets of stakeholders for which the property is valid or consistent, respectively. The stakeholders correspond to join-irreducibles in our framework. The reduction described in [HP03] results in $|C|$ single-view partial models. Verification on each model is performed by switching between “valid” and “consistent” interpretations of satisfiability of properties.

5 Conclusion

In this paper, we studied the problem of multi-valued μ-calculus model-checking. Instead of solving the problem directly, we reduced it to several classical problems that can be solved using existing model-checking tools. The number of such problems depends on the number of join-irreducible elements of the logic, and each problem is linear in the size of the property and the model. We have also put numerous existing work in the area of non-classical verification into the context of our work.

Our results enable construction of clever algorithms that use results obtained from classical problems and the order of join-irreducibles to minimize the number of redundant checks. Yet, optimality can be achieved only by solving these problems in parallel, which is done by "true" multi-valued model-checkers such as χChek [CDG02].

Acknowledgment. We thank Michael Huth, Benet Devereux and the anonymous referees for helping improve the presentation of this paper. Financial support for this research has been provided by NSERC and CITO.

References

[BG99] G. Bruns and P. Godefroid. "Model Checking Partial State Spaces with 3-Valued Temporal Logics". In *Proceedings of 11th International Conference on Computer-Aided Verification (CAV'99)*, volume 1633 of *Lecture Notes in Computer Science*, pages 274–287, Trento, Italy, 1999. Springer.

[BG00] G. Bruns and P. Godefroid. "Generalized Model Checking: Reasoning about Partial State Spaces". In C. Palamidessi, editor, *Proceedings of 11th International Conference on Concurrency Theory (CONCUR'00)*, volume 1877 of *Lecture Notes in Computer Science*, pages 168–182, University Park, PA, USA, August 2000. Springer.

[Bry86] R. E. Bryant. "Graph-based algorithms for boolean function manipulation.". *Transactions on Computers*, 8(C-35):677–691, 1986.

[CDE+01] M. Chechik, B. Devereux, S. Easterbrook, A. Lai, and V. Petrovykh. "Efficient Multiple-Valued Model-Checking Using Lattice Representations". In K.G. Larsen and M. Nielsen, editors, *Proceedings of 12th International Conference on Concurrency Theory (CONCUR'01)*, volume 2154 of *Lecture Notes in Computer Science*, pages 451–465, Aalborg, Denmark, August 2001. Springer.

[CDEG03] M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. "Multi-Valued Symbolic Model-Checking". *ACM Transactions on Software Engineering and Methodology*, 2003. (Accepted for publication.).

[CDG02] M. Chechik, B. Devereux, and A. Gurfinkel. "χChek: A Multi-Valued Model-Checker". In *Proceedings of 14th International Conference on Computer-Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 505–509, Copenhagen, Denmark, July 2002. Springer.

[CEP01] M. Chechik, S. Easterbrook, and V. Petrovykh. "Model-Checking Over Multi-Valued Logics". In *Proceedings of Formal Methods Europe (FME'01)*, volume 2021 of *Lecture Notes in Computer Science*, pages 72–98. Springer, March 2001.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications". *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.

[CGP99] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[Cha00] W. Chan. "Temporal-Logic Queries". In E. Emerson and A. Sistla, editors, *Proceedings of the 12th Conference on Computer Aided Verification (CAV'00)*, volume 1855 of *Lecture Notes in Computer Science*, pages 450–463, Chicago, IL, USA, July 2000.

[DP90] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[Eme90] E.A. Emerson. "Temporal and Modal Logic". In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, Amsterdam, 1990.

[Fit69] M.C. Fitting. *Intuitionistic Logic Model Theory and Forcing*. North-Holand Publishing Co., Amsterdam, 1969.

[Fit91] M. Fitting. "Many-Valued Modal Logics". *Fundamenta Informaticae*, 15(3-4):335–350, 1991.

[Fit92] M. Fitting. "Many-Valued Modal Logics II". *Fundamenta Informaticae*, 17:55–73, 1992.

[GCD03] A. Gurfinkel, M. Chechik, and B. Devereux. "Temporal Logic Query Checking: A Tool for Model Exploration". *IEEE Transactions on Software Engineering*, 2003. (Accepted for publication.).

[GHJ01] P. Godefroid, M. Huth, and R. Jagadeesan. "Abstraction-based Model Checking using Modal Transition Systems". In K.G. Larsen and M. Nielsen, editors, *Proceedings of 12th International Conference on Concurrency Theory (CONCUR'01)*, volume 2154 of *Lecture Notes in Computer Science*, pages 426–440, Aalborg, Denmark, 2001. Springer.

[GJ03] P. Godefroid and R. Jagadeesan. "On the Expressiveness of 3-Valued Models". In *Proceedings of 4th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'03)*, volume 2575 of *Lecture Notes in Computer Science*, pages 206–222, New York, USA, January 2003. Springer.

[HJS01] M. Huth, R. Jagadeesan, and D. A. Schmidt. "Modal Transition Systems: A Foundation for Three-Valued Program Analysis". In *Proceedings of 10th European Symposium on Programming (ESOP'01)*, volume 2028 of *Lecture Notes in Computer Science*, pages 155–169. Springer, 2001.

[HJS03] M. Huth, R. Jagadeesan, and D. Schmidt. "A Domain Equation for Refinement of Partial Systems". *Mathematical Structures in Computer Science*, 2003. (Accepted for publication).

[HP03] M. Huth and S. Pradhan. "An Ontology for Consistent Partial Model Checking". *Electronic Notes in Theoretical Computer Science*, 23, 2003.

[Kle52] S. C. Kleene. *Introduction to Metamathematics*. New York: Van Nostrand, 1952.

[Koz83] D Kozen. "Results on the Propositional μ-calculus". *Theoretical Computer Science*, 27:334–354, 1983.

[KP02a] B. Konikowska and W. Penczek. "Model Checking for Multi-Valued CTL*". In M. Fitting and E. Orlowska, editors, *Multi-Valued Logics*. 2002. (to appear).

[KP02b] B. Konikowska and W. Penczek. "Reducing Model Checking from Multi-Valued CTL* to CTL*". In *Proceedings of 13 International Conference on Concurrency Theory (CONCUR'02)*, volume 2421 of *Lecture Notes in Computer Science*, Brno, Czech Republic, August 2002. Springer.

[LT88] K.G. Larsen and B. Thomsen. "A Modal Process Logic". In *Proceedings of 3rd Annual Symposium on Logic in Computer Science (LICS'88)*, pages 203–210. IEEE Computer Society Press, 1988.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, New York, 1989.

[MOSS99] M. Müller-Olm, D. Schmidt, and B. Steffen. "Model-Checking: A Tutorial Introduction". In *Proceedings of 6th International Static Analysis Symposium (SAS'99*, volume 1694 of *Lecture Notes in Computer Science*, pages 330–354, Berlin, September 1999.

[Ras78] H. Rasiowa. *An Algebraic Approach to Non-Classical Logics. Studies in Logic and the Foundations of Mathematics*. Amsterdam: North-Holland, 1978.

A Proof of Theorem 5

Theorem 5. *Let φ be a $L_\mu(\mathcal{L})$ formula, $j \in \mathcal{J}(\mathcal{L})$, and s be a state of a χKripke structure. Then there exists a $L_\mu^{\sqsupseteq}$ formula $\varphi \Uparrow j$, called the* cut *of φ with respect to j, such that $||\varphi||_\rho(s) \sqsupseteq j \;=\; ||\varphi \Uparrow j||_\rho(s)$.*

Proof:
The proof proceeds by induction on the structure of φ. We only provide the proofs for the existential and universal next-time operators. The proofs for the rest of the operators are similar.

For the existential next-time operator $\Diamond$, we get:

$$\begin{array}{lr}
\ ||\Diamond\varphi||_\rho(s) \sqsupseteq j & \text{(Definition of } ||\cdot||_\rho\text{)} \\
= \bigsqcup_{t\in S}(\mathbb{R}(s,t) \wedge ||\varphi||_\rho(t)) \sqsupseteq j & \text{(distributivity of } \cdot \sqsupseteq j\text{)} \\
= \bigsqcup_{t\in S}(\mathbb{R}(s,t) \sqsupseteq j \wedge ||\varphi||_\rho(t) \sqsupseteq j) & \text{(induction hypothesis)} \\
= \bigsqcup_{t\in S}(\mathbb{R}(s,t) \sqsupseteq j \wedge ||\varphi \Uparrow j||_\rho(t)) & \text{(Definition of } ||\cdot||_\rho\text{)} \\
= ||\langle\sqsupseteq j\rangle(\varphi \Uparrow j)||_\rho(s) &
\end{array}$$

Finally, for the universal next-time operator $\Box$, we get:

$$\begin{array}{lr}
\ ||\Box\varphi||_\rho(s) \sqsupseteq j & \text{(Definition of } ||\cdot||_\rho\text{)} \\
= \sqcap_{t\in S}(\mathbb{R}(s,t) \Rightarrow ||\varphi||_\rho(t)) \sqsupseteq j & \text{(Definition of } \Rightarrow\text{)} \\
= \sqcap_{t\in S}(\neg\mathbb{R}(s,t) \vee ||\varphi||_\rho(t)) \sqsupseteq j & \text{(distributivity of } \cdot \sqsupseteq j\text{)} \\
= \sqcap_{t\in S}((\neg\mathbb{R}(s,t)) \sqsupseteq j \vee ||\varphi||_\rho(t) \sqsupseteq j) & \text{(induction hypothesis)} \\
= \sqcap_{t\in S}((\neg\mathbb{R}(s,t)) \sqsupseteq j \vee ||\varphi \Uparrow j||_\rho(t)) & \text{(Theorem 2)} \\
= \sqcap_{t\in S}(\neg(\mathbb{R}(s,t) \sqsupseteq \mathsf{neg}(j)) \vee ||\varphi \Uparrow j||_\rho(t)) & \text{(Definition of } ||\cdot||_\rho\text{)} \\
= ||[\sqsupseteq \mathsf{neg}(j)](\varphi \Uparrow j)||_\rho(s) &
\end{array}$$

$\square$