Systematic Construction of Abstractions for Model-Checking

Arie Gurfinkel, Ou Wei, and Marsha Chechik

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

This paper describes a framework, based on Abstract Interpretation, for creating abstractions for model-checking. Specifically, we study how to abstract models of μ-calculus and systematically derive abstractions that are constructive, sound, and precise, and apply them to abstracting Kripke structures. The overall approach is based on the use of bilattices to represent partial and inconsistent information.

1 Introduction

Abstraction plays a fundamental role in combating state-space explosion in model-checking. The goal of abstraction is to construct an abstract model of a system which is small enough to be effectively analyzed, and yet rich enough to yield conclusive results. Success of current abstraction projects, such as SLAM [2] and Bandera [6], indicates that abstraction is an effective technique for enabling model-checking of realistic software systems.

In model-checking, a notion of abstracting a transition system is typically developed as follows: (1) An abstract statespace is defined such that each abstract state corresponds to a set of concrete states. This correspondence can be arbitrary, as in predicate abstraction [17], or influenced by the concrete statespace, as in symmetry reduction [12]. (2) An abstract transition system is constructed by defining a transition relation over this abstract statespace. (3) Finally, the resulting system is argued to be sound, i.e., it is shown to preserve a fragment of the desired temporal logic.

The problem with the above approach is that it is not algorithmic: the techniques used to construct the abstract systems require a certain amount of intuition of users, and extra effort is needed to show that the resulting abstraction is correct. This makes it difficult to understand a specific abstraction method and improve on it. For example, given an abstraction that preserves universal CTL, how should it be changed to preserve the entire CTL? It is also difficult to understand the relationship between different abstract methods. For example, as shown in [25], predicate abstraction and symmetry reduction differ only in their choice of abstract states. However, this insight was not apparent just from the description of these methods.

Given the role abstraction plays in the model-checking process, we believe it is essential to create a general methodology for systematically constructing and analyzing abstractions. In the context of static analysis of programs, such a framework, called Abstract Interpretation (AI), has already been proposed by [7]. It provides a collection of notations and tools to formalize the approximation of program semantics, as well as to design and analyze program abstractions. The goal of this paper is to specialize the AI framework to model-checking.

There are a number of ways to do this specialization, given the breadth of model-checking approaches. Our goal here is to create abstractions that preserve properties expressed in the modal μ-calculus [20] (L_μ). Following the recipes of AI, we systematically derive conditions under which an abstract L_μ model is the best abstraction of a concrete one. We guarantee that these abstract models are (a) sound, i.e., if an L_μ formula is satisfied in the abstract model it is satisfied in the concrete, (b) most precise, i.e., satisfy the most properties, and (c) have the desired structural characteristics, e.g., a requirement that an abstraction of a transition system is a transition system as well. These conditions are constructive and, as we show in this paper, can be derived almost mechanically. The algorithm for building a desired abstraction follows from these conditions directly.

The logic L_μ includes negation, so that an L_μ formula $\neg\varphi$ is satisfied iff φ is refuted. If we assume that every formula is either satisfied or refuted in an abstraction as well, it may seem that preserving soundness for *all* L_μ formulas means that such an abstraction must satisfy and refute exactly the same properties as the corresponding concrete model (resulting in a bisimilar model). If the goal is to save space for model-checking, this abstraction would be very limited. Thus, most existing abstractions are restricted to fragments of L_μ, i.e., only to the universal or only to the existential properties (see, e.g., [22]).

The insight we use in this paper is that an abstraction is inherently incomplete: some formulas may be neither satisfied nor refuted by it. We propose to treat satisfaction and refutation independently. If we classify all L_μ formulas using a pair $\langle \mathit{Sat}, \mathit{Ref} \rangle$, where *Sat* contains all the formulas satisfied in an abstract model, while *Ref* contains all the refuted ones, then *Sat* and *Ref* are not necessarily complements of each other. In fact, *Sat* and *Ref* may not even be disjoint, allowing some formulas to be both satisfied and refuted.

Associating knowledge about truth and falsity of every piece of evidence can be naturally encoded using 4-valued Belnap logic [3] which enjoys nice mathematical properties associated with *bilattices* [15, 14]. That is, bilattices enable a uniform approach for handling partial and inconsistent information, allowing reasoning about truth and knowledge in a single theoretical framework. In this paper, by combining the theories of AI with that of bilattices, we obtain a simple and elegant framework for deriving abstractions for L_μ. Due to the generality of bilattices, our results apply not only to the traditional two-valued interpretation of L_μ, but also to its multi-valued [4] and quantitative [10] interpretations.

The contribution of this paper is a general technique, based on AI, for deriving abstractions for model-checking. It allows understanding and comparing different techniques, and provides a methodology for proving soundness and precision of the desired abstraction. We then study this technique on two additional levels. First, we apply it to L_μ models, and then specialize it to abstracting transition systems represented as Kripke structures.

The rest of this paper is organized as follows: after providing the necessary background in Section 2, we show how to lift abstraction between elements to abstraction between sets of elements in Section 3. This gives us a general framework for approximating interpretations of L_μ. In Section 4, we derive abstractions for model-theoretic interpretations of L_μ, and then apply our technique to abstracting transition systems in Section 5. In Section 6, we specialize the results of Section 5 to *boolean* transition systems, and compare them to those obtained by Dams et al. [8]. We relate our technique with other abstraction approaches in Section 7 and summarize our contributions in Section 8.

2 Background

In this section, we introduce the basic concepts of lattice theory, modal μ-calculus, and abstract interpretation.

2.1 Lattices and Monotone Functions

A *lattice* is a partially ordered set $\mathcal{L} = (L, \leq)$ in which every subset B of L has a least upper bound, called *join* and denoted $\sqcup B$, and a greatest lower bound, called *meet* and denoted $\sqcap B$ [9]. A lattice is *distributive* if meet and join distribute over each other, i.e., $(a \sqcap b) \sqcup c = (a \sqcup c) \sqcap (b \sqcup c)$, and $(a \sqcup b) \sqcap c = (a \sqcap c) \sqcup (b \sqcap c)$.

A *De Morgan algebra* is a structure $\mathcal{D} = (L, \leq, -)$, where $(L, \leq)$ is a distributive lattice and $- : L \to L$ is a *negation* that satisfies involution ($- - a = a$) and De Morgan laws: $-(a \sqcap b) = -a \sqcup -b$, and $-(a \sqcup b) = -a \sqcap -b$.

We denote the space of functions from A to B by $A \to B$, or B^A. For example, both $A \to [B \to C]$ and $(C^B)^A$ denote the space of functions from A to functions from B to C.

Let A be a set and $\mathcal{L} = (L, \leq)$ be a lattice. The ordering and operations of $\mathcal{L}$ extend pointwise to L^A, i.e., $f \leq g \Leftrightarrow \forall a \in A \cdot f(a) \leq g(a)$. This turns L^A into a lattice with the same properties as $\mathcal{L}$. In particular, if $\mathcal{L}$ is distributive or De Morgan, so is L^A.

A function f between two partially ordered sets $(A, \leq)$ and $(B, \sqsubseteq)$ is *monotone* (or, *order-preserving*) iff $a \leq b \Rightarrow f(a) \sqsubseteq f(b)$, and *anti-monotone* iff $a \leq b \Rightarrow f(b) \sqsubseteq f(a)$. We use upward ($\uparrow$) and downward ($\downarrow$) arrows to indicate monotone and anti-monotone functions, respectively. For example, $[A \to B]_\uparrow$ denotes the space of all monotone functions from A to B, and $(B^A)_\downarrow$ denotes the space of all anti-monotone functions. Monotone and anti-monotone functions are closed under pointwise meet and join; thus, if B is a lattice, then so are $[A \to B]_\uparrow$ and $[A \to B]_\downarrow$.

2.2 Truth Domains and Sets

A *truth-domain* $\mathcal{D}$ is a collection of elements D, referred to as *truth values*, together with a truth ordering $\sqsubseteq$ and a negation operator $\neg : D \to D$, such that $\mathcal{D} = (D, \sqsubseteq, \neg)$ is a De Morgan algebra. The truth ordering orders the elements based on their truth content; thus, $a \sqsubseteq b$ stands for "a is less true than b". The meet ($\wedge$) and join ($\vee$) of the truth ordering are called *conjunction* and *disjunction*, respectively.

The best known truth domain is the classical boolean logic $\mathbf{2}$ with values true and false. Its truth ordering is shown in the Hasse diagram in Figure 1(a), with negation indicated in parentheses. Other examples include Belnap logic $\mathcal{B}$, shown in Figure 1(b), which extends boolean logic with two additional values: m and d, to represent "unknown" and "inconsistent", respectively; and Fuzzy logic $\mathcal{F}$, shown in Figure 1(c). The truth values of $\mathcal{F}$ are formed by the set of all real numbers in the closed interval $[0, 1]$, where 0 stands for false, 1 for true, and the remaining values stand for degrees of truth; furthermore, negation is defined as $\neg x \triangleq 1 - x$, so that $\neg 0 = 1$ and $\neg 1 = 0$.

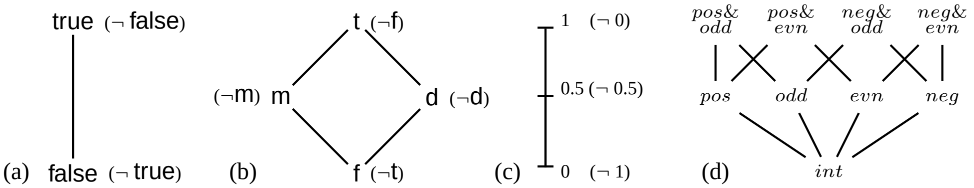

Fig. 1. (a)-(c) Truth domains: (a) 2-valued boolean logic, (b) Belnap logic, and (c) Fuzzy logic. (d) An abstract domain for $\mathbb{Z}$.

Given a collection of elements U, a *set* over U is a function from U to a truth domain. Thus, a boolean (or classical) set is a function from U to $\mathbf{2}$, a Belnap set is a function from U to $\mathcal{B}$, and a fuzzy set is a function from U to $\mathcal{F}$. Set ordering and operations are defined by pointwise extensions. Let $S_1, S_2 : U \to D$ be two sets. Then

$$\begin{array}{llll} S_1 \subseteq S_2 & \triangleq \forall x \cdot S_1(x) \sqsubseteq S_2(x) & \qquad S_1 \cup S_2 & \triangleq \lambda x \cdot S_1(x) \vee S_2(x) \\ \overline{S_1} & \triangleq \lambda x \cdot \neg S_1(x) & \qquad S_1 \cap S_2 & \triangleq \lambda x \cdot S_1(x) \wedge S_2(x) \end{array}$$

2.3 Modal μ-Calculus

In this section, we describe the modal μ-calculus [20], or L_μ.

Definition 1. *Let* Var *be a set of variables and* AP *be a set of atomic propositions. The logic* $L_\mu(AP)$ *is the set of all formulas satisfying the grammar*

$$\varphi ::= p \mid z \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \mu z \cdot \varphi(z),$$

where $p \in AP$*, and* $z \in \mathsf{Var}$*. Furthermore,* z *in* $\mu z \cdot \varphi(z)$ *must occur under the scope of an even number of negations in* $\varphi(z)$*.*

Additionally, we define the following syntactic abbreviations:

$$\varphi \vee \psi \triangleq \neg(\neg\varphi \wedge \neg\psi) \quad \varphi \Rightarrow \psi \triangleq \neg\varphi \vee \psi \quad \Box\varphi \triangleq \neg\Diamond\neg\varphi \quad \nu Z \cdot \varphi(Z) \triangleq \neg\mu Z \cdot \neg\varphi(\neg Z)$$

The modal operator $\Diamond$ is typically interpreted as “an existence of an immediate future”. For example, “p” means that p holds now, “$\Diamond p$” means that there exists an immediate future where p holds, and “$\Box p$” means that p holds in all immediate futures. The quantifiers μ and ν stand for least and greatest fixpoint, respectively.

An occurrence of a variable z in a formula φ is *bound* if it appears in the scope of a μ quantifier and is *free* otherwise. For example, z is free in $p \vee \Diamond z$, and is bound in $\mu z \cdot p \vee \Diamond z$. A formula φ is *closed* if it does not contain any free variables.

A *set-based interpretation* of L_μ over a set domain $\mathcal{D}^C$ is a mapping $||\cdot||$ from closed L_μ formulas to $\mathcal{D}$-sets over C. The elements of C are often called *states*, and $||\varphi||(c) = v$ is read as “the degree to which φ is satisfied by (or, true in) a state c is v”.

An L_μ *model* is a structure $\mathcal{M} = (\mathcal{D}^C, (p^{\mathcal{M}})_{p \in AP}, \Diamond^{\mathcal{M}})$, where $\mathcal{D}^C$ is a set domain; for each $p \in AP$, $p^{\mathcal{M}}$ is in $\mathcal{D}^C$; and $\Diamond^{\mathcal{M}} : \mathcal{D}^C \to \mathcal{D}^C$ is a $\subseteq$-monotone function. The set domain is called the *universe* of $\mathcal{M}$, and $p^{\mathcal{M}}$ and $\Diamond^{\mathcal{M}}$ are interpretations of atomic propositions and the $\Diamond$ operator, respectively. A model $\mathcal{M}$ gives rise to an L_μ interpretation $||\cdot||^{\mathcal{M}}$.

The interpretation $||\varphi||^{\mathcal{M}}_\sigma$ is defined inductively on the structure of the formula φ, where $\sigma : \mathsf{Var} \to \mathcal{D}^C$ is an object assignment for free variables:

$$\begin{array}{llll} ||p||^{\mathcal{M}}_\sigma & \triangleq p^{\mathcal{M}} & \qquad ||z||^{\mathcal{M}}_\sigma & \triangleq \sigma(z) \\ ||\varphi \wedge \psi||^{\mathcal{M}}_\sigma & \triangleq ||\varphi||^{\mathcal{M}}_\sigma \cap ||\psi||^{\mathcal{M}}_\sigma & \qquad ||\neg\varphi||^{\mathcal{M}}_\sigma & \triangleq \overline{||\varphi||^{\mathcal{M}}_\sigma} \\ ||\mu x \cdot \varphi||^{\mathcal{M}}_\sigma & \triangleq \mathrm{lfp}^{\subseteq}\left(\lambda S \cdot ||\varphi||^{\mathcal{M}}_{\sigma[x \mapsto S]}\right) & \qquad ||\Diamond\varphi||^{\mathcal{M}}_\sigma & \triangleq \Diamond^{\mathcal{M}}(||\varphi||^{\mathcal{M}}_\sigma) \end{array}$$

where $\text{lfp}^{\subseteq} f$ is the $\subseteq$-least fixpoint of f. For a closed L_μ formula φ, $||\varphi||^{\mathcal{M}}_{\sigma} = ||\varphi||^{\mathcal{M}}_{\sigma'}$ for any σ and σ'. Thus, we write $||\varphi||^{\mathcal{M}}$ for that value, and define it to be the *model-based interpretation* of φ.

Formulas of L_μ are often interpreted over Kripke structures. A *Kripke structure* is a tuple $\mathcal{K} = (AP, C, \mathcal{D}, I, R)$, where AP is a collection of atomic propositions, C is a collection of elements (called *states*), $\mathcal{D}$ is a truth domain, $I : AP \to \mathcal{D}^C$ is a mapping from atomic propositions to sets over C, and $R : C \to \mathcal{D}^C$ is a transition function mapping each state to a set of its successors. For a transition function R, we define a pre-image operator $\textit{pre}[R] : \mathcal{D}^C \to \mathcal{D}^C$ and its dual $\tilde{\textit{pre}}[R]$ as:

$$\textit{pre}[R](Q)(s) \triangleq \vee_{t \in C}(R(s) \cap Q)(t) \qquad \tilde{\textit{pre}}[R](Q) \triangleq \overline{\textit{pre}[R](\overline{Q})}$$

Intuitively, $\textit{pre}[R](Q)(s)$ is a degree to which the set $R(s)$ of successors of s has a non-empty intersection with Q. A Kripke structure $\mathcal{K} = (AP, C, \mathcal{D}, I, R)$ gives rise to an $L_\mu(AP)$ model $\mathcal{M}(\mathcal{K}) = (\mathcal{D}^C, (p^{\mathcal{M}(\mathcal{K})})_{p \in AP}, \Diamond^{\mathcal{M}(\mathcal{K})})$, where $p^{\mathcal{M}(\mathcal{K})} \triangleq I(p)$, and $\Diamond^{\mathcal{M}(\mathcal{K})} \triangleq \textit{pre}[R]$. Finally, the interpretation $|| \cdot ||^{\mathcal{K}}$ of L_μ in $\mathcal{K}$ is defined as $||\varphi||^{\mathcal{K}} \triangleq ||\varphi||^{\mathcal{M}(\mathcal{K})}$.

2.4 Abstract Interpretation

The framework of Abstract Interpretation (AI) provides a collection of tools for systematic design and analysis of semantic approximations [7]. The framework is very flexible and can be applied in various ways. Below, we give a brief overview of AI, summarizing the results used in our work.

Basics of Abstract Interpretation. Inputs to an AI framework are collections of concrete elements C and abstract elements A, called a *concrete* and an *abstract* domain, respectively. A notion of approximation, or abstraction, is formalized by a *soundness relation* $\rho \subseteq A \times C$, where $a \,\rho\, c$ is read as "a ρ-approximates c".

A *concretization function* $\gamma : A \to 2^C$ maps each abstract element to a set of concrete elements corresponding to it: $\gamma(a) \triangleq \{c \mid a \,\rho\, c\}$. An abstract element a is called *consistent* if $\gamma(a) \neq \emptyset$; otherwise, we say a is *inconsistent*. The elements of A can be thought of as properties, such as "positive" or "odd", and $\gamma(a)$ as a collection of concrete elements satisfying a. The concretization γ induces an *approximation ordering* $\preceq_\rho$ on A such that $a \preceq_\rho b \Leftrightarrow \gamma(a) \supseteq \gamma(b)$.

Intuitively, $a \preceq_\rho b$ means that a approximates more concrete elements than b; therefore, a is less informative, or equivalently, less precise than b. When viewed as a property, a is weaker than b. For example, knowing that an element is "positive" is less informative than knowing that it is both "positive" and "odd".

In this paper, an abstract domain A is equipped with an *information ordering* $\preceq_A$ such that $(A, \preceq_A)$ is a lattice and $a \preceq_A b \Rightarrow a \preceq_\rho b$. Thus, we can study properties of an abstract domain independently of any particular soundness relation. Furthermore, we assume that A satisfies "the existence of a best approximation" [7], that is:

$$\forall c \in C \cdot \exists a \in A \cdot (a \,\rho\, c \wedge \forall a' \in A \cdot a' \,\rho\, c \Rightarrow \gamma(a') \supseteq \gamma(a))$$

and use $\alpha : C \to A$ to denote an *abstraction function* that maps each concrete element to its best approximation. Note that for a given c, A may have several best approximations; thus, α is not uniquely defined. In such cases, it is convenient to use the $\preceq_A$-largest α, so that ρ and γ can be expressed as $a\,\rho\,c \Leftrightarrow a \preceq_A \alpha(c)$ and $c \in \gamma(a) \Leftrightarrow a \preceq_A \alpha(c)$, respectively.

A lower bound with respect to $\preceq_\rho$ is called *widening* and is denoted by $\triangledown$. Intuitively, for a set $Q \subseteq A$, $\triangledown Q$ is an abstract element representing the information common to all elements of Q, i.e., $\gamma(\triangledown Q) \supseteq \cup_{q \in Q} \gamma(q)$. In particular, the greatest lower bound $\sqcap_A$ of $\preceq_A$ is a widening. A widening $\triangledown$ is *info-preserving* if for any Q containing no inconsistent elements, $\triangledown Q$ is the best representation of information common to all elements of Q, i.e., $\forall a \in A \cdot \gamma(a) \supseteq \cup_{q \in Q} \gamma(q) \Rightarrow \gamma(a) \supseteq \gamma(\triangledown Q)$.

Abstract domains $(A_1, \preceq_1)$ and $(A_2, \preceq_2)$ are *informationally equivalent* if they represent the same degrees of information, that is, $\forall a_1 \in A_1 \cdot \exists a_2 \in A_2 \cdot \gamma_1(a_1) = \gamma_2(a_2)$ and $\forall a_2 \in A_2 \cdot \exists a_1 \in A_1 \cdot \gamma_1(a_1) = \gamma_2(a_2)$.

For examples in this paper, we use the set of integers $\mathbb{Z}$ as a concrete domain, and the domain $\mathbb{A}$, shown in Figure 1(d), as the abstract domain. The soundness relation $\rho_e \subseteq \mathbb{A} \times \mathbb{Z}$ is self-explanatory, e.g., 2 is ρ_e-approximated by $pos\&evn$, evn, pos, and int, where $pos\&evn$ is its best abstract approximation. Similarly, $\gamma_e(evn)$ is the set *EVEN* of all even numbers, $\gamma_e(neg)$ is the set *NEG* of all negative numbers, $\gamma_e(neg\&evn)$ is *NEG* $\cap$ *EVEN*, etc.

Functional Abstraction. In practice, it is common to synthesize abstractions of complex structures using abstractions of their parts. A particular application is abstraction of functions, or *functional abstraction* [7].

Let $\mathcal{A} = A_1 \to A_2$ and $\mathcal{C} = C_1 \to C_2$ be collections of abstract and concrete functions, where A_1 and A_2 are abstract domains approximating C_1 and C_2, respectively. A soundness relation $\rho_f \subseteq \mathcal{A} \times \mathcal{C}$ is *functional* if g ρ_f-approximates f iff g preserves soundness of f. Formally, ρ_f satisfies

$$g\,\rho_f\,f \Leftrightarrow \forall a_1 \in A_1 \cdot \forall c_1 \in \gamma_1(a_1) \cdot g(a_1)\,\rho_2\,f(c_1) \qquad \text{(functional soundness)}$$

Let $\triangledown$ be a widening operator of A_2, and $\alpha_\triangledown : \mathcal{C} \to \mathcal{A}$ be defined as

$$\alpha_\triangledown(f)(a) \triangleq \triangledown_{c \in \gamma_1(a)} \alpha_2(f(c)) \qquad \text{(functional abstraction)}$$

Then $\alpha_\triangledown(f)$ is a ρ_f-approximation of f, and its precision is determined by the precision of the widening operator used.

Theorem 1. *[7] Let $\mathcal{A}, \mathcal{C}, \rho_f$, and $\alpha_\triangledown$ be as above. If $\triangledown$ is info-preserving, then $\alpha_\triangledown(f)$ is the best ρ_f-approximation of f.*

One of the main results of AI is that $\alpha_\triangledown$ preserves fixpoints:

Theorem 2. *[7] Let $(C, \sqsubseteq_C)$ be a lattice, $f : [C \to C]_\uparrow$ be a monotone function, and $(A, \sqsubseteq_A)$ be a lattice approximating C via ρ_C. If the join operator $\vee_A$ of A preserves soundness, i.e., $(\alpha_C(c_1) \vee_A \alpha_C(c_2)) \preceq_A \alpha_C(c_1 \vee_C c_2)$, then the least fixpoint of $\alpha_\triangledown(f)$ ρ_C-approximates the least fixpoint of f: $\mathrm{lfp}^{\sqsubseteq_A} \alpha_\triangledown(f) \;\; \rho_C \;\; \mathrm{lfp}^{\sqsubseteq_C} f$.*

Functional Abstraction and Monotone Functions. Let $\mathcal{A} = [A_1 \to A_2]$ be as above, and assume that A_1 and A_2 are equipped with information orderings $\preceq_1$ and $\preceq_2$, respectively. Then the set $\mathcal{A}_\uparrow = [A_1 \to A_2]_\uparrow$ of $\preceq$-monotone functions is informationally equivalent to $\mathcal{A}$.Furthermore, if $\triangledown$ is an info-preserving widening of A_2, then its pointwise extension to functions is also an info-preserving widening of $\mathcal{A}_\uparrow$ [25]. Therefore, we always restrict the abstract domain of functional abstraction to $\preceq$-monotone functions.

3 Abstract Sets

Sets play the role of basic blocks in the definition of L_μ semantics. In this section, we develop an abstraction of sets that preserves all set operations, including set complement. This abstraction gives us the necessary tools for abstracting L_μ models, which we do in Section 4. But it is independent of L_μ and can be used anywhere abstract sets are required.

We assume that C and A are a concrete and an abstract domain, respectively, related by a soundness relation ρ_e and an abstraction function α_e. We aim to lift ρ_e to a soundness relation ρ_s between concrete sets, i.e., functions from C into a fixed truth domain $\mathcal{D}$, and abstract sets, i.e., functions from A into some truth domain $\mathcal{B}$ (potentially different from $\mathcal{D}$). The goal of ρ_s is to preserve set membership: that is, if S_α ρ_s-approximates S, then if $a \in A$ ρ_e-approximates c, $S_\alpha(a)$ must approximate $S(c)$. As always, we also want to know when S_α is a best approximation of a given set S.

We view sets as functions, so it is natural to express ρ_s as a functional abstraction. For this, we must first identify the notion of an abstract truth domain $\mathcal{B}$ and settle on the meaning of "approximating truth values".

3.1 Bilattices as Abstract Truth Domains

Intuitively, an abstract truth-domain $\mathcal{B}$ is a truth-domain and, therefore, has a truth ordering and a negation. It is also an abstract domain and needs an information ordering. Furthermore, truth operations should not interfere with the information ordering. For example, if a and b are in $\mathcal{B}$ and a is less informative than b, then negation of a ($\neg a$) must be less informative than $\neg b$.

A structure that captures our intuition is that of a bilattice, which has been introduced by Ginsberg [15] to enable reasoning with partiality and inconsistency. Here, we briefly describe distributive bilattices.

Definition 2. *[15] A* distributive bilattice *is a structure $\mathcal{B} = (B, \preceq, \sqsubseteq, \neg)$ such that: (1) $\mathcal{B}_i = (B, \preceq)$ is a lattice and $\mathcal{B}_t = (B, \sqsubseteq, \neg)$ is a De Morgan algebra; (2) meet ($\sqcap$) and join ($\sqcup$) of $\mathcal{B}_i$, and meet ($\wedge$) and join ($\vee$) of $\mathcal{B}_t$ are monotone with respect to both $\preceq$ and $\sqsubseteq$; (3) all meets and joins distribute over each other; and (4) negation ($\neg$) is $\preceq$-monotone.*

The ordering $\preceq$ ranks elements of $\mathcal{B}$ with respect to information, and $\sqsubseteq$ ranks them with respect to truth. Operations $\wedge$ and $\vee$ of $\mathcal{B}_t$ are called conjunction and disjunction. In the spirit of AI, we refer to $\sqcap$ and $\sqcup$ as *widening* and *narrowing*, respectively.

De Morgan algebras have a natural connection to bilattices.

Theorem 3. *[15, 14]. Let $\mathcal{D} = (D, \leq, -)$ be a De Morgan algebra, and $\mathcal{B}(\mathcal{D})$ be a structure $(D \times D, \preceq, \sqsubseteq, \neg)$ such that*

$$\langle a, b\rangle \preceq \langle c, d\rangle \triangleq a \leq c \wedge b \leq d \qquad \langle a, b\rangle \sqsubseteq \langle c, d\rangle \triangleq a \leq c \wedge d \leq b \qquad \neg\langle a, b\rangle \triangleq \langle b, a\rangle$$

Then, $\mathcal{B}(\mathcal{D})$ is a distributive bilattice. Furthermore, every distributive bilattice is isomorphic to $\mathcal{B}(\mathcal{D})$ for some De Morgan algebra $\mathcal{D}$.

For a truth-domain $\mathcal{D}$, an element $\langle x, y\rangle$ of $\mathcal{B}(\mathcal{D})$ is interpreted as a truth value whose degree of truth is x and degree of falsity is y. For example, $\mathcal{B}(\mathbf{2})$ consists of four elements: $\langle \mathsf{t}, \mathsf{f}\rangle$ representing true – maximal degree of truth and minimal degree of falsity, $\langle \mathsf{f}, \mathsf{t} \rangle$ representing false, $\langle \mathsf{f}, \mathsf{f} \rangle$ representing lack of knowledge – minimal degree of both truth and falsity, and $\langle \mathsf{t}, \mathsf{t} \rangle$ representing an inconsistency (or disagreement) – maximal degree of both truth and falsity. It is easy to verify that $\mathcal{B}(\mathbf{2})$ is exactly Belnap logic shown in Figure 1(b). For convenience, we introduce projections π_{t} and π_{f} defined as $\pi_{\mathsf{t}}(\langle x, y \rangle) \triangleq x$ and $\pi_{\mathsf{f}}(\langle x, y \rangle) \triangleq y$.

Guided by the above intuition, we say that $\mathcal{B}(\mathcal{D})$ is an *abstract truth-domain* corresponding to a truth domain $\mathcal{D}$. Intuitively, $\langle x, y \rangle \in \mathcal{B}(\mathcal{D})$ approximates $c \in \mathcal{D}$ if x is no more true than c, and y is no more false than c. In particular, $\langle c, -c \rangle$ is the best approximation of c. Formally, this is captured by an abstraction function $\alpha_t(c) \triangleq \langle c, -c \rangle$, and a soundness relation $\rho_t \triangleq \{(a, c) \mid a \preceq \alpha_t(c)\}$.

It is easy to verify that truth operations of $\mathcal{B}(\mathcal{D})$, including negation, preserve soundness. That is, if $a_1 \preceq \alpha_t(c_1)$ and $a_2 \preceq \alpha_t(c_2)$, then $a_1 \wedge a_2 \preceq \alpha_t(c_1 \wedge c_2)$, $a_1 \vee a_2 \preceq \alpha_t(c_1 \vee c_2)$, and $\neg a_1 \preceq \alpha_t(\neg c_1)$. Furthermore, $\sqcap$ is an info-preserving widening.

3.2 Set Abstraction

We now formally define the soundness relation ρ_s between concrete ($C \to \mathcal{D}$) and abstract ($A \to \mathcal{B}(\mathcal{D})$) sets as:

$$S_\alpha \, \rho_s \, S \triangleq \forall a \in A \cdot \forall c \in \gamma_e(a) \cdot S_\alpha(a) \preceq \alpha_t(S(c)) \qquad \text{(set soundness)}$$

The soundness relation ρ_s is functional, and the corresponding abstraction function α_s follows immediately from Theorem 1:

$$\alpha_s(S)(a) \triangleq \sqcap_{c \in \gamma_e(a)} \alpha_t(S(c)) \qquad \text{(set abstraction)}$$

Note that $\alpha_s(S)(a) = \langle x, y \rangle$ means that the elements in $\gamma_e(a)$ belong to S with a truth degree of at least x, and to $\overline{S}$ with a truth degree of at least y. In particular, if S is a boolean set, then $\alpha_s(S)$ is a Belnap set; $\alpha_s(S)(a)$ is t iff $\gamma_e(a)$ is contained in S, f iff $\gamma_e(a)$ is contained in $\overline{S}$, m iff $\gamma_e(a)$ is not contained in either S or $\overline{S}$, and d iff $\gamma_e(a)$ is contained in both S and $\overline{S}$.

For example, an abstraction $\alpha_s(\mathit{EVEN})$ of a boolean set $\mathit{EVEN} \in \mathbf{2}^{\mathbb{Z}}$ is

$$\alpha_s(\mathit{EVEN})(a) \triangleq \begin{cases} \mathsf{t} & \text{if } \gamma_e(a) \subseteq \mathit{EVEN} \\ \mathsf{f} & \text{if } \gamma_e(a) \subseteq \mathit{ODD} \\ \mathsf{m} & \text{otherwise} \end{cases}$$

Note a difference between an abstract element evn and an abstract set $\alpha_s(\mathit{EVEN})$. The former represents a property of being an even number, and $\gamma_e(evn) = \mathit{EVEN}$ is the set of all numbers having this property. On the other hand, $\alpha_s(\mathit{EVEN})$ represents a set that contains all even and no odd numbers; hence, $\gamma_s(\alpha_s(\mathit{EVEN})) = \{\mathit{EVEN}\}$ is a singleton containing the only set satisfying these conditions.

Recall that the set operations of $\mathcal{B}(\mathcal{D})^A$ are pointwise extensions of the corresponding operations of $\mathcal{B}(\mathcal{D})$; therefore, they preserve soundness. For example, if S_α ρ_s-approximates S, then $\overline{S_\alpha}$ ρ_s-approximates $\overline{S}$, etc.

Finally, since ρ_s is functional, following the discussion in Section 2.4, we restrict the domain of abstract sets to $\preceq$-monotone functions, i.e., to $\mathcal{B}(\mathcal{D})^A_{\uparrow}$. Note that abstract set operations preserve $\preceq$-monotonicity and do not interfere with this restriction. This gives us with a abstract domain for sets that (a) preserves all set operations and (b) has an info-preserving widening. We use elements of this abstract domain as basic blocks for designing L_μ-preserving abstractions in the next section.

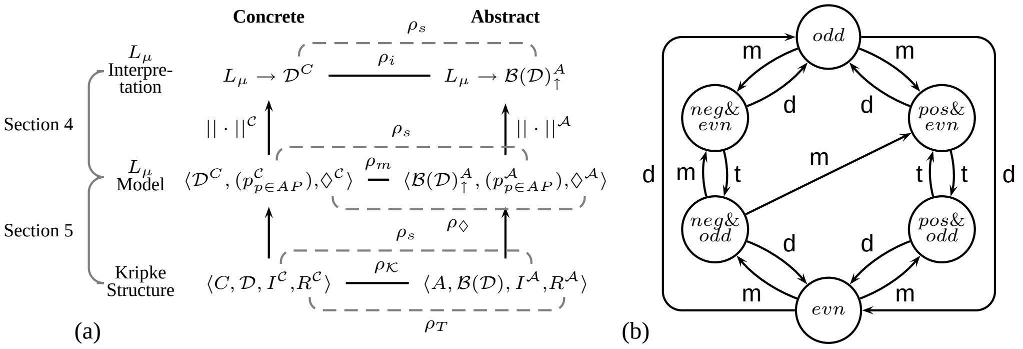
Concrete
Abstract
ρ_s
L_μ Interpretation
$L_\mu \to \mathcal{D}^C$
ρ_i
$L_\mu \to \mathcal{B}(\mathcal{D})^A_\uparrow$
Section 4
$||\cdot||^\mathcal{C}$
ρ_s
$||\cdot||^\mathcal{A}$
L_μ Model
$\langle \mathcal{D}^C, (p^\mathcal{C}_{p\in AP}), \Diamond^\mathcal{C}\rangle$
ρ_m
$\langle \mathcal{B}(\mathcal{D})^A_\uparrow, (p^\mathcal{A}_{p\in AP}), \Diamond^\mathcal{A}\rangle$
Section 5
ρ_s
$\rho_\Diamond$
Kripke Structure
$\langle C, \mathcal{D}, I^\mathcal{C}, R^\mathcal{C}\rangle$
$\rho_\mathcal{K}$
$\langle A, \mathcal{B}(\mathcal{D}), I^\mathcal{A}, R^\mathcal{A}\rangle$
ρ_T
(a)
(b)
odd
neg& evn
pos& evn
neg& odd
pos& odd
evn
m
d
t

Fig. 2. (a) Abstracting L_μ: the top row summarizes soundness relations for abstracting $L\mu$ interpretations; the middle one – L_μ models, i.e., interpretions of atomic propositions and $\Diamond$ relation; the bottom one – L_μ-preserving abstractions of Kripke structures. (b) A fragment of the abstraction $\alpha_T(R_1)$, where $R_1(x) = x + 1$.

4 Abstract Interpretation for Modal μ-Calculus

In this section, we develop an abstraction of L_μ models that is sound w.r.t. satisfaction and refutation of all L_μ formulas, i.e., if an L_μ formula is satisfied (refuted) by the abstract model, it is satisfied (refuted) by the concrete one. We start by formalizing the notion of L_μ-preserving approximation in the language of AI, and then systematically extend it to the desired abstraction. The top half of the diagram in Figure 2(a) illustrates the structures and relations discussed in this section, where solid lines represent relations between structures, and dashed those between their components.

We assume that C is a collection of concrete elements, called states, and $\mathcal{D}$ is a truth domain. Recall from Section 2.3 that an interpretation of L_μ $||\cdot||$ over a set domain $\mathcal{D}^C$ maps each closed L_μ formula to a $\mathcal{D}$-set over C, where $||\varphi||(c)$ is the degree to which a formula φ is true in a state c.

Let A be an abstract domain approximating C via a soundness relation ρ_e, and $\mathcal{B}(\mathcal{D})$ be an abstract truth domain approximating $\mathcal{D}$ via a soundness relation ρ_t as defined in Section 3.1. Furthermore, let $||\cdot||_\alpha$ be an interpretation of L_μ formulas as $\mathcal{B}(\mathcal{D})$-sets over A.

A natural way to extend the soundness relation ρ_e from states to L_μ interpretations is to say that $||\cdot||_\alpha$ approximates $||\cdot||$ if for every L_μ formula φ and every abstract state $a \in A$, $||\varphi||_\alpha(a)$ approximates the degree to which $||\varphi||$ is true for every concrete state c corresponding to a. We denote this soundness relation by ρ_i and formalize it using the set soundness relation ρ_s, defined in Section 3.2, as

$$||\cdot||_\alpha \;\rho_i\; ||\cdot|| \triangleq \forall \varphi \in L_\mu \cdot ||\varphi||_\alpha \;\rho_s\; ||\varphi|| \qquad (L_\mu \text{ soundness})$$

In this paper, we are only interested in the model-based interpretations of L_μ. A natural way to extend ρ_i to models is to say that a concrete model $\mathcal{C}$ is approximated by an abstract model $\mathcal{A}$ if the corresponding L_μ interpretation $||\cdot||^\mathcal{C}$ is approximated by $||\cdot||^\mathcal{A}$. Formally, we define a model soundness relation ρ_m as

$$\mathcal{A}\,\rho_m\,\mathcal{C} \triangleq ||\cdot||^\mathcal{A} \;\rho_i\; ||\cdot||^\mathcal{C} \qquad \text{(model soundness)}$$

In the rest of this section, we employ the AI framework to construct an abstract model $\mathcal{A}$ that is a best ρ_m-approximation of a given concrete model $\mathcal{C}$. As discussed in Section 3, we restrict the universe of $\mathcal{A}$ to $\preceq$-monotone functions from A to $\mathcal{B}(\mathcal{D})$.

We first outline the steps involved: (1) define a soundness relation $\rho_\Diamond$ between interpretations of the $\Diamond$ operator and derive the corresponding abstraction function $\alpha_\Diamond$; (2) show that an abstract model $\mathcal{A} = (\mathcal{B}(\mathcal{D})^A_\uparrow, (p^\mathcal{A})_{p \in AP}, \Diamond^\mathcal{A})$ ρ_m-approximates a concrete model $\mathcal{C} = (\mathcal{D}^C, (p^\mathcal{C})_{p \in AP}, \Diamond^\mathcal{C})$ if for each $p \in AP$, $p^\mathcal{A}$ ρ_s-approximates $p^\mathcal{C}$, and $\Diamond^\mathcal{A}$ $\rho_\Diamond$-approximates $\Diamond^\mathcal{C}$; (3) conclude that the best approximation of $\mathcal{C}$ is given by $\alpha_m(\mathcal{C}) \triangleq (\mathcal{B}(\mathcal{D})^A_\uparrow, (\alpha_s(p^\mathcal{C}))_{p \in AP}, \alpha_\Diamond(\Diamond^\mathcal{C}))$.

Step 1. For a given L_μ-model, an interpretation of modal formulas, i.e. formulas with $\Diamond$ but no fixpoint quantifiers, is determined by the model's interpretation of the $\Diamond$ operator. Thus, we define $\rho_\Diamond$ as follows:

$$\Diamond^\mathcal{A} \ \rho_\Diamond \ \Diamond^\mathcal{C} \triangleq \forall X \in \mathcal{B}(\mathcal{D})^A_\uparrow \cdot \forall Y \in \gamma_s(X) \cdot \Diamond^\mathcal{A}(X) \ \rho_s \ \Diamond^\mathcal{C}(Y) \qquad (\Diamond\text{-soundness})$$

Following Theorem 1, its corresponding abstraction function $\alpha_\Diamond$ is defined as

$$\alpha_\Diamond(\Diamond^\mathcal{C})(X) \triangleq \sqcap_{Y \in \gamma_s(X)} \alpha_s(\Diamond^\mathcal{C}(Y)) \qquad (\Diamond\text{-abstraction})$$

Step 2. To show that an abstract model $\mathcal{A}$ ρ_m-approximates a concrete model $\mathcal{C}$ if each component of $\mathcal{A}$ approximates the corresponding counterpart of $\mathcal{C}$, we need to show that for any formula φ, $||\varphi||^\mathcal{A}$ ρ_s-approximates $||\varphi||^\mathcal{C}$.

Theorem 4. *Let $\mathcal{C} = (\mathcal{D}^C, (p^\mathcal{C})_{p \in AP}, \Diamond^\mathcal{C})$ be a concrete model, $\mathcal{A} = (\mathcal{B}(\mathcal{D})^A, (p^\mathcal{A})_{p \in AP}, \Diamond^\mathcal{A})$ be an abstract model such that A approximates C via a soundness relation ρ_e. Then, $\mathcal{A} \, \rho_m \, \mathcal{C} \Leftarrow \forall p \in AP \cdot p^\mathcal{A} \, \rho_s \, p^\mathcal{C} \wedge \Diamond^\mathcal{A} \, \rho_\Diamond \, \Diamond^\mathcal{C}$.*

The theorem is proved by structural induction on φ, using Theorem 2 for cases where φ contains a fixpoint quantifier.

Step 3. Finally, we define an abstraction function α_m that maps each concrete model to its best abstract approximation:

$$\alpha_m(\mathcal{C}) \triangleq (\mathcal{B}(\mathcal{D})^A_\uparrow, (\alpha_s(p^\mathcal{C}))_{p \in AP}, \alpha_\Diamond(\Diamond^\mathcal{C})) \qquad (\text{model abstraction})$$

For example, consider a concrete boolean model $\mathcal{C} = (\mathbf{2}^\mathbb{Z}, p^\mathcal{C}, \Diamond^\mathcal{C})$, where $p^\mathcal{C} = \mathit{EVEN}$ and $\Diamond^\mathcal{C} = \lambda S \cdot \{y \mid y + 1 \in S\}$. Then, $\Diamond p$ is interpreted in $\mathcal{C}$ as $||\Diamond p||^\mathcal{C} = \Diamond^\mathcal{C}(\mathit{EVEN}) = \mathit{ODD}$, and in the abstraction of $\mathcal{C}$ as $||\Diamond p||^{\alpha_m(\mathcal{C})} = \alpha_\Diamond(\Diamond^\mathcal{C})(\alpha_s(\mathit{EVEN})) = \alpha_s(\mathit{ODD})$.

The resulting abstraction function α_m allows us to abstract L_μ models, obtaining abstractions which are both sound and precise. However, α_m depends on an interpretation of $\Diamond$ modality, which we left unspecified. We study this subject below.

5 Abstraction of Kripke Structures

In practice, the $\Diamond$ modality is often interpreted using a Kripke structure. In this section, we are interested in conditions under which a Kripke structure over an abstract statespace (i.e., an abstract Kripke structure) is a best approximation of a given concrete one. We show that the framework of AI provides an elegant and almost mechanical way to answer this question.

Approximating Kripke Structures. Below, we aim to extend the soundness relation ρ_m between models to a soundness relation $\rho_\mathcal{K}$ between Kripke structures, and derive a corresponding abstraction function $\alpha_\mathcal{K}$.

Throughout this section, we assume that $\mathcal{C} = (C, \mathcal{D}, I^\mathcal{C}, R^\mathcal{C})$ is a concrete Kripke structure over concrete states C and a truth domain $\mathcal{D}$, and $\mathcal{A} = (A, \mathcal{B}(\mathcal{D}), I^\mathcal{A}, R^\mathcal{A})$

$$\begin{tikzcd}
\mathcal{D}^C \to \mathcal{D}^C \arrow[r, dash, "\rho_\Diamond"] & \mathcal{B}(\mathcal{D})^A_\uparrow \to \mathcal{B}(\mathcal{D})^A_\uparrow \\
C \to [\mathcal{D}^C \to \mathcal{D}] \arrow[u] \arrow[r, phantom, dashed, bend left, "\rho_{\mathrm{isct}}"] & A \to [\mathcal{B}(\mathcal{D})^A_\uparrow \to \mathcal{B}(\mathcal{D})]_\uparrow \arrow[u] \\
R^{\mathcal{C}} = C \to \mathcal{D}^C \arrow[u] \arrow[uu, bend left=60, "pre[R^{\mathcal{C}}]"] \arrow[r, dash, "\rho_T"] \arrow[r, dashed, dash, bend left, "\rho_\cap"] & R^{\mathcal{A}} = A \to \mathcal{B}(\mathcal{D})^A_\downarrow \arrow[u] \arrow[uu, bend right=60, "pre[R^{\mathcal{A}}]"']
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
{[\mathcal{D}^C \to \mathcal{D}]} \arrow[r, dash, "\rho_{\mathrm{isct}}"] & {[\mathcal{B}(\mathcal{D})^A_\uparrow \to \mathcal{B}(\mathcal{D})]_\uparrow} \\
\mathcal{D}^C \arrow[u, "\mathrm{isct}(X)"'] \arrow[r, dash, "\rho_\cap"] & \mathcal{B}(\mathcal{D})^A_\downarrow \arrow[u, "\mathrm{isct}(Y)"']
\end{tikzcd}$$

(b)

Fig. 3. (a) Soundness relations between $\Diamond$ modality and transition function; (b) Detail of (a): relations ρ_{isct} and $\rho_\cap$.

is an abstract Kripke structure, where A is an abstract domain related to C via ρ_e, and $\mathcal{B}(\mathcal{D})$ is an abstract truth domain related to $\mathcal{D}$ via ρ_t.

The soundness relation $\rho_{\mathcal{K}}$ on Kripke structures is defined as a restriction of the model soundness relation ρ_m (see Figure 2(a)):

$$\mathcal{A} \, \rho_{\mathcal{K}} \, \mathcal{C} \triangleq \mathcal{M}(\mathcal{A}) \, \rho_m \, \mathcal{M}(\mathcal{C}) \qquad \text{(Kripke soundness)}$$

By Theorem 4, $\rho_{\mathcal{K}}$ is decomposed over the components of the Kripke structure:

$$\mathcal{A} \, \rho_{\mathcal{K}} \, \mathcal{C} \Leftarrow (\forall p \in AP \cdot I^{\mathcal{A}}(p) \, \rho_s \, I^{\mathcal{C}}(p)) \wedge R^{\mathcal{A}} \, \rho_T \, R^{\mathcal{C}}$$

where the relation ρ_T between transition functions is defined as:

$$R^{\mathcal{A}} \, \rho_T \, R^{\mathcal{C}} \triangleq \mathit{pre}[R^{\mathcal{A}}] \, \rho_\Diamond \, \mathit{pre}[R^{\mathcal{C}}] \qquad \text{(transition soundness)}$$

The abstraction function α_s corresponding to ρ_s has already been defined in Section 3.2. Thus, the only missing ingredient for defining $\alpha_{\mathcal{K}}$ is the transition abstraction α_T. Unfortunately, the soundness relation ρ_T is not functional; making Theorem 1 not applicable. However, we show below that ρ_T can be easily made functional. We begin by introducing an intersection operator isct: $\mathrm{isct}(X)(S) \triangleq \vee_t (X \cap S)(t)$ which allows us to express the pre-image of a transition function R as $\mathit{pre}[R](Q) = \lambda s \cdot \mathrm{isct}(R(s))(Q)$. We then define a functional soundness relation ρ_{isct} (see Figure 3(a)):

$$\mathrm{isct}(X) \, \rho_{\mathrm{isct}} \, \mathrm{isct}(Y) \triangleq \forall S \in \mathcal{B}(\mathcal{D})^A_\uparrow \cdot \forall Q \in \gamma_s(S) \cdot \mathrm{isct}(X)(S) \;\; \rho_t \;\; \mathrm{isct}(Y)(Q)$$

Noticing that $\mathrm{isct}(X)$ is determined by a set X, we extend ρ_{isct} to a soundness relation $\rho_\cap$ between sets (see Figure 3(b)):

$$X \, \rho_\cap \, Y \triangleq \mathrm{isct}(X) \, \rho_{\mathrm{isct}} \, \mathrm{isct}(Y) \qquad \text{(successor soundness)}$$

Finally, ρ_T is made functional:

$$\begin{aligned}
R^{\mathcal{A}} \, \rho_T \, R^{\mathcal{C}} &\Leftrightarrow \mathit{pre}[R^{\mathcal{A}}] \, \rho_\Diamond \, \mathit{pre}[R^{\mathcal{C}}] \\
&\Leftrightarrow \forall a \in A \cdot \forall c \in \gamma_e(s) \cdot \mathrm{isct}(R^{\mathcal{A}}(a)) \, \rho_{\mathrm{isct}} \, \mathrm{isct}(R^{\mathcal{C}}(c)) \\
&\Leftrightarrow \forall a \in A \cdot \forall c \in \gamma_e(s) \cdot R^{\mathcal{A}}(a) \, \rho_\cap \, R^{\mathcal{C}}(c)
\end{aligned}$$

However, $\rho_\cap$ is still not functional! Thus, before applying Theorem 1 to construct α_T, we need to construct the abstraction function $\alpha_\cap$ directly, i.e., without using Theorem 1. We do so below.

Abstraction of Intersection. Intuitively, the ideal abstraction $\alpha_\cap$ is such that the diagram in Figure 3(b) commutes. That is, $\alpha_\cap(X) = Y$ implies that $\alpha_{\mathrm{isct}}(\mathrm{isct}(X)) = \mathrm{isct}(Y)$. Note that ρ_{isct} is functional, thus the definition of $\alpha_{\mathrm{isct}}(\mathrm{isct}(X))$ follows from Theorem 1. Following a standard technique of AI, we proceed to reorganize this definition until the emergence of conditions under which $Y \in \mathcal{B}(\mathcal{D})^A$ is the best $\rho_\cap$-abstraction of X. This derivation is simple but long, and is omitted from the paper. For details, please see full version of this paper [19]. Here, we only show the final result.

Theorem 5. *Let C and $(A, \preceq_A)$ be a concrete and an abstract domain related by ρ_e, $\mathcal{D}$ and $\mathcal{B}(\mathcal{D})$ be truth-domains related by ρ_t, and for $X \in \mathcal{D}^A$, let $\alpha_\cap$ be defined as*

$$\alpha_\cap(X)(a) \triangleq \langle \vee_{c \in \gamma_e(a)} X(c), \wedge_{c \in \tilde{\gamma}_e(a)} \neg X(c) \rangle,$$

where $\tilde{\gamma}_e(a) \triangleq \{c \in C \mid \alpha_e(c) \preceq_A a\}$ is a dual-conretization function. Then, $\alpha_{\text{isct}}(\text{isct}(X)) = \text{isct}(\alpha_\cap(X))$.

To construct α_T using Theorem 1, we need an info-preserving widening. The widening $\sqcap$ on $\mathcal{B}(D)^A$ – the pointwise extension of $\sqcap$ of $\mathcal{B}(\mathcal{D})$ – is not info-preserving in general. Instead, we restrict the abstract domain to the $\preceq$-antimonotone functions, i.e., to $\mathcal{B}(\mathcal{D})^A_\downarrow$, since (a) $\mathcal{B}(\mathcal{D})^A_\downarrow$ is informationally equivalent to $\mathcal{B}(\mathcal{D})^A$, and (b) it makes pointwise widening $\sqcap$ info-preserving. Note that $\alpha_\cap(X)$ is already $\preceq$-antimonotone.

Abstraction of Transition Functions. Once the abstraction $\alpha_\cap$ is defined, the abstraction of transition functions α_T follows from Theorem 1:

$$\alpha_T(R^{\mathcal{C}})(a) \triangleq \sqcap_{c \in \gamma_e(a)} \alpha_\cap(R^{\mathcal{C}}(c)) \qquad \text{(transition abstraction)}$$

By expanding $\alpha_\cap$, α_T can be alternatively expressed as:

$$\pi_{\mathsf{t}}(\alpha_T(R^{\mathcal{C}})(a)(b)) = \wedge_{c \in \gamma_e(a)}\ \textit{pre}[R^{\mathcal{C}}](\gamma_e(b))(c)$$
$$\pi_{\mathsf{f}}(\alpha_T(R^{\mathcal{C}})(a)(b)) = \wedge_{c \in \gamma_e(a)}\ \tilde{\textit{pre}}[R^{\mathcal{C}}](\neg\tilde{\gamma}_e(b))(c)$$

That is, if $R^{\mathcal{A}} = \alpha_T(R^{\mathcal{C}})$, then a transition $R^{\mathcal{A}}(a)(b)$ between abstract states a and b is as true as the least degree with which all concrete states in $\gamma_e(a)$ have a successor in $\gamma_e(b)$, and as false as the least degree with which all successors of states in $\gamma_e(a)$ are *not* in $\tilde{\gamma}_e(b)$.

Note that so far, we have made no assumptions on the concrete transition function. However, if the concrete transition function $R^{\mathcal{C}}$ is boolean, then $R^{\mathcal{A}} = \alpha_T(R^{\mathcal{C}})$ is $\mathcal{B}(\mathbf{2})$-valued and satisfies:

$$R^{\mathcal{A}}(a)(b) = \langle \gamma_e(a) \subseteq \textit{pre}[R^{\mathcal{C}}](\gamma_e(b)), \gamma_e(a) \subseteq \tilde{\textit{pre}}[R^{\mathcal{C}}](\neg\tilde{\gamma}_e(b)) \rangle$$

For example, let $R_1(x) \triangleq x + 1$. A fragment of its abstraction $\alpha_T(R_1)$ is shown in Figure 2(b), where pos, neg and int are removed for clarity. For any even x, $x + 1$ is *definitely* odd, but it *maybe* positive or negative. Thus, the transition from evn to odd is d, and transitions to $pos\&odd$ and to $neg\&odd$ are m. Note that the pre-image of $\alpha_T(R_1)$ approximates the pre-image of R_1, e.g., $\textit{pre}[\alpha_T(R_1)](\alpha_s(\textit{EVEN})) = \alpha_s(\textit{ODD})$.

Finally, the best abstract Kripke structure $\alpha_{\mathcal{K}}(\mathcal{C})$ of a concrete Kripke structure $\mathcal{C} = (C, \mathcal{D}, I^{\mathcal{C}}, R^{\mathcal{C}})$ is obtained compositionally:

$$\alpha_{\mathcal{K}}(\mathcal{C}) \triangleq (A, \mathcal{B}(\mathcal{D}), \alpha_s \circ I^{\mathcal{C}}, \alpha_T(R^{\mathcal{C}})) \qquad \text{(Kripke abstraction)}$$

Thus, we were able to systematically derive rules for abstracting Kripke structures by abstract Kripke structures.

Note that the diagram in Figure 3(a) does not commute, i.e., $\alpha_\diamond(\textit{pre}[R]) \neq \textit{pre}[\alpha_T(R)]$. Thus, for a given Kripke structure, its best abstraction by an abstract L_μ-model is more precise than its best abstraction by an abstract Kripke structure. For example, let R_2 be

$$R_2(x) \triangleq \begin{cases} 2x & \text{if } x \geq 5 \wedge x \in \textit{ODD} \\ -x & \text{if } 0 \leq x < 5 \wedge x \in \textit{ODD} \\ -2 & \text{otherwise} \end{cases}$$

and $X \triangleq (\mathit{POS} \cap \mathit{EVEN}) \cup (\mathit{NEG} \cap \mathit{ODD})$. Then, $\alpha_\Diamond(\mathit{pre}[R_2])(\alpha_s(X))(pos\&odd) = \alpha_s(\mathit{POS} \cap \mathit{ODD})(pos\&odd) = \mathsf{t}$, but $\mathit{pre}[\alpha_T(R_2)](\alpha_s(X))(pos\&odd) = \mathsf{m}$. This shows that transition systems are not necessarily the best abstract domain for L_μ-preserving abstractions.

6 Application: Abstraction of Classical Kripke Structures

In this section, we look at boolean Kripke structures and compare our abstraction to that of Dams et al. [8], which provides an alternative way of computing the best L_μ-preserving abstraction of Kripke structures.

We begin by addressing minor differences between the two approaches. First, the goal of [8] is to preserve satisfaction of positive L_μ, i.e., a fragment of L_μ with negation restricted to atomic propositions. Second, Kripke structures are abstracted by Mixed Transition Systems (MixTSs). Essentially, a MixTS is a Kripke structure with two separate transition relations, R^C and R^F, called *constrained* and *free*, respectively. The interpretation of L_μ over MixTSs is the same as its interpretation over Kripke structures, with the exception that $\Diamond$ is interpreted as $\mathit{pre}[R^C]$ and $\Box$ – as $\widetilde{\mathit{pre}}[R^F]$.

Note that positive L_μ is as expressive as full L_μ: for every L_μ formula φ there exists an equivalent positive formula $\mathit{NNF}(\varphi)$, its negation normal form. Thus, an abstraction that preserves positive L_μ easily extends to full L_μ. Furthermore, the next theorem shows that MixTSs are equivalent to $\mathcal{B}(\mathbf{2})$-valued Kripke structures.

Theorem 6. *Let $\mathcal{T}$ be a MixTS with statespace A and transition functions R^C and R^F, and $\mathcal{K}$ be a $\mathcal{B}(\mathbf{2})$-valued Kripke structure with the same statespace, and a transition function $R^{\mathcal{K}}$ such that $R^{\mathcal{K}}(a)(b) = \langle R^C(a)(b), \neg R^F(a)(b)\rangle$. Then, for any L_μ formula φ, $||\varphi||^{\mathcal{K}} = \langle ||\mathit{NNF}(\varphi)||^{\mathcal{T}}, ||\mathit{NNF}(\neg\varphi)||^{\mathcal{T}}\rangle$.*

Thus, in the case of boolean Kripke structures, the abstraction developed in this paper is equivalent to that of [8]: same structures are used as an abstract domain, and exactly the same L_μ formulas are preserved. However, unlike the approach taken in [8], our work systematically derives both the abstraction and the notion of abstract Kripke structures from L_μ-preservation and the soundness relation ρ_s between concrete and abstract sets.

It is interesting to note that although the two abstractions are equivalent w.r.t satisfaction of L_μ, they are not identical. For completeness, Dams et al. show that the most precise MixTS abstracting a Kripke structure satisfies the following conditions:

$$\begin{array}{c} R^C(a,b) \Leftrightarrow b \in \{\sqcap_{y \in Y} \alpha_e(y) \mid Y \in \min\{Y' \mid R^{\forall\exists}(\gamma_e(a), Y')\}\} \\ R^F(a,b) \Leftrightarrow b \in \{\sqcap_{y \in Y} \alpha_e(y) \mid Y \in \min\{Y' \mid R^{\exists\exists}(\gamma_e(a), Y')\}\} \end{array}$$

where $R^{\forall\exists}(S,T) \triangleq \forall s \in S \cdot \exists t \in T \cdot R(s)(t)$ and $R^{\exists\exists}(S,T) \triangleq \exists s \in S \cdot \exists t \in T \cdot R(s)(t)$. It is different from our abstraction α_T, which, when put in this notation, is:

$$\alpha_T(R)(a)(b) = \langle R^{\forall\exists}(\gamma_e(a), \gamma_e(b)), \neg R^{\exists\exists}(\gamma_e(a), \tilde{\gamma}_e(b))\rangle$$

We believe that our characterization is simpler; however, it remains to be seen whether it is also more useful in practice, e.g., if it leads to a smaller symbolic representation, or easier to construct compositionally, etc. We leave this topic for future work.

7 Related Work

Over the years, many abstraction methods have been developed for L_μ model-checking [5, 8, 12, 17, 21, 22, 24]. They concentrate on a specific model – transition systems and most of them preserve soundness (satisfaction) for fragments of L_μ: if an abstract system is an over-approximation of the concrete one, the abstraction is sound for all universal properties. Similarly, a sound abstraction for existential properties comes from under-approximation.

The first approach for sound abstraction of *full* L_μ was proposed by Larsen and Thompsen [21]. They have shown that Modal Transition Systems (MTS) can be used to combine both over- and under-approximations. However, the goal of this work is not abstraction, and it did not consider the problem of how to abstract a Kripke structure using an MTS. The construction problem is addressed by Dams et al. [8], who independently proposed using MixTSs, a slight generalization of MTSs, as abstract models, and provided conditions for constructing an MixTS with the best precision. Although this work uses AI to describe the relationship between concrete and abstract statespaces, abstract transition systems are not derived systematically; instead, the optimal conditions are defined based on intuition, and both soundness and optimality of precision require separate proofs.

Among the attempts of using AI to systematically derive best abstractions, the work of Loiseaux et al. [22] and Schmidt [23] are the closest to ours. [22] showed how to derive a simulation-based sound abstract transition system from Galois connections within the AI framework, but their results apply only to the universal fragment of L_μ. Motivated by the study of MixTSs, [23] showed how to capture over- and under-approximations between transition systems using AI and systematically derived Dams's most precise results. However, the starting goal of this work was formalizing the over- and the under-approximations, restricting the result to the specific L_μ models, namely, transition systems. On the other hand, in our work we start from formalizing the notion of soundness of L_μ interpretations – the most general and exact goal of abstraction for L_μ (via the soundness relation ρ_i in Section 4), and then systematically derive conditions which guarantee the best precision of the abstraction. Thus, our results can be applied to different L_μ models, where abstracting transition systems is just a special case.

Another important feature of our work is the use of bilattices. The approaches of [8, 23] develop best over- and under-approximations separately, whereas our combination of AI with bilattices provides a uniform way for abstraction of both satisfaction and refutation of L_μ. Multi-valued logic has been previously combined with abstraction in the form of 3-valued transition systems (e.g. [16]). However, these results do not use the framework of AI, and, in particular, only deal with soundness and not the precision of the abstraction. Furthermore, 3-valued Kripke structures (unlike those based on Belnap logic) lack monotonicity [24]: a more refined abstract domain does not necessarily result in a more precise abstraction, and thus the most precise abstraction may not even exist.

8 Conclusion

In this paper, we have shown that abstract interpretation provides a systematic way for designing abstractions for model-checking. On one hand, our work can be seen as recreating the pioneering work of Dams et al. [8] in a systematic setting where each step in designing an abstraction and each loss of precision can be traced back to either the choice of an abstract domain, or the requirements on the abstract structure. On the other hand, our work also extends their results to non-traditional interpretations of L_μ, such as its multi-valued [4] and quantitative [10] interpretations. To the best of our knowledge, this is the first abstraction technique that can be applied to these non-classical interpretations.

In this paper, we lay the basic groundwork for designing L_μ-preserving abstractions using the framework of AI. However, our work can be easily extended in a number of directions. We discuss a few of them below.

We have shown that requiring that an abstraction of a transition system be a transition system as well, comes with a loss of precision. Thus, it may be interesting to explore how a transition system can be abstracted directly by an abstract L_μ model. Such models will require new model-checking algorithms, but will provide additional precision, and possibly be easier to construct. For example, recent work on symmetry reduction [13] argues that instead of constructing a reduced abstract model, the symmetry-reduced $\Diamond$ modality can be implemented directly by putting symmetry reduction inside the model-checking algorithm. We believe that our framework can be used to extend this approach to other, non-symmetry induced, abstract domains. Our work on a software model-checker YASM [18] is a first step in this direction.

In designing abstractions of Kripke structures, we have assumed that the domain and range of the transition function are abstracted by the same abstract domain. This need not be the case. By using different but related abstract domains, we obtain a generalization of “hyper-transition abstractions” [24, 11] to arbitrary abstract domains.

Although not shown explicitly in the paper, the pointwise extension of the bilattice narrowing operator $\sqcup$ to abstract structures provides a simple way to combine several, not necessarily best, abstractions. This allows us to study incremental construction of abstractions, such as the one in [1].

We believe that our framework provides an interesting starting point for exploring the connection between AI and model-checking, and hope to continue this line of research in the future.

References

1. T. Ball, V. Levin, and F. Xie. “Automatic Creation of Environment Models via Training”. In *TACAS’04*, volume 2988 of *LNCS*, pages 93–107, 2004.
2. T. Ball and S. Rajamani. “The SLAM Toolkit”. In *CAV’01*, volume 2102 of *LNCS*, pages 260–264, 2001.
3. N.D. Belnap. “A Useful Four-Valued Logic”. In Dunn and Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 30–56. Reidel, 1977.
4. M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. “Multi-Valued Symbolic Model-Checking”. *ACM TOSEM*, 12(4):1–38, 2003.
5. Edmund M. Clarke, Orna Grumberg, and David E. Long. “Model Checking and Abstraction”. *ACM TOPLAS*, 16(5):1512–1542, 1994.
6. J. Corbett, M. Dwyer, J. Hatcliff, S. Laubach, C. Pasareanu, Robby, and H. Zheng. “Bandera: Extracting Finite-state Models from Java Source Code”. In *ICSE’00*, pages 439–448, 2000.
7. P. Cousot and R. Cousot. “Abstract Interpretation Frameworks”. *Journal of Logic and Computation*, 2(4):511–547, 1992.

8. D. Dams, R. Gerth, and O. Grumberg. "Abstract Interpretation of Reactive Systems". *ACM TOPLAS*, 2(19):253–291, 1997.
9. B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.
10. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. "Model Checking Discounted Temporal Properties". In *TACAS'04*, volume 2988 of *LNCS*, pages 77–92, 2004.
11. L. de Alfaro, P. Godefroid, and R. Jagadeesan. "Three-Valued Abstractions of Games: Uncertainty, but with Precision". In *LICS'04*, pages 170–179, 2004.
12. E. A. Emerson and A. P. Sistla. "Symmetry and Model Checking". *FMSD*, 9(1-2):105–131, 1996.
13. E. A. Emerson and T. Wahl. "Dynamic Symmetry Reduction". In *TACAS'05*, volume 3440 of *LNCS*, pages 382–396, 2005.
14. M. Fitting. "Bilattices are Nice Things". In *Conference on Self-Reference*, 2002.
15. M. L. Ginsberg. "Multivalued Logics: A Uniform Approach to Reasoning in Artificial Intelligence". *Computational Intelligence*, 4(3):265–316, 1988.
16. P. Godefroid, M. Huth, and R. Jagadeesan. "Abstraction-based Model Checking using Modal Transition Systems". In *CONCUR'01*, volume 2154 of *LNCS*, pages 426–440, 2001.
17. S. Graf and H. Saïdi. "Construction of Abstract State Graphs with PVS". In *CAV'97*, volume 1254 of *LNCS*, pages 72–83, 1997.
18. A. Gurfinkel and M. Chechik. "Yasm: Model-Checking Software with Belnap Logic". Technical Report 533, University of Toronto, April 2005.
19. A. Gurfinkel, O. Wei, and M. Chechik. "Logical Abstract Interpretation". Technical Report 532, University of Toronto, September 2005.
20. D Kozen. "Results on the Propositional μ-calculus". *Theoretical Computer Science*, 27:334–354, 1983.
21. K.G. Larsen and B. Thomsen. "A Modal Process Logic". In *LICS'88*, pages 203–210, 1988.
22. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. "Property Preserving Abstractions for the Verification of Concurrent Systems". *FMSD*, 6:1–35, 1995.
23. D. A. Schmidt. "Closed and Logical Relations for Over- and Under-Approximation of Powersets". In *SAS'04*, volume 3148 of *LNCS*, pages 22–37, 2004.
24. S. Shoham and O. Grumberg. "Monotonic Abstraction-Refinement for CTL". In *TACAS'04*, LNCS, pages 546–560, April 2004.
25. O. Wei, A. Gurfinkel, and M. Chechik. "Identification and Counter Abstraction for Full Virtual Symmetry". In *CHARME'05*, volume 3725 of *LNCS*, 2005.