CTL Model-Checking over Logics with Non-Classical Negations

Marsha Chechik
Department of Computer Science
University of Toronto
Toronto, ON, Canada M5S 2E4
chechik@cs.toronto.edu

Wendy MacCaull
Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University
Antigonish, NS, Canada B2G 2W5
wmaccaul@stfx.ca

Abstract

In earlier work [9], we defined CTL model-checking over finite-valued logics with De Morgan negation. In this paper, we extend this work to logics with intuitionistic, Galois and minimal negations, calling the resulting language χCTL. We define χCTL operators and show that they can be computed using fixpoints. We further discuss how to extend our existing multi-valued model-checker χChek [8] to reasoning over these logics.

1 Introduction

Logics with non-classical negation have a number of applications in modeling and reasoning about systems. For example, intuitionistic logic can be used for program analysis and optimization [2] and for investigating safety properties of behaviours of reactive systems [1]. Such logics provide a setting for the study of composition and refinement rules, and a framework for the use of the modular specification methods. Further, De Morgan logics with finite number of elements can be used for aggregating information coming from different sources [14], for reasoning about partial systems [5, 16], for compiler optimization [24], and for query-checking [15].

In [6, 7], Chechik et al. introduced model-checking over quasi-boolean logics. They built χChek [8] – a multi-valued symbolic model-checker which generalizes an existing symbolic model-checking algorithm to allow reasoning over a multi-valued extension of CTL (χCTL). Given a system expressed as a χKripke structure – a multi-valued extension of a classical Kripke structure – and a χCTL property, χChek returns the *degree* to which the system satisfies the property. The original work considered those logics where the truth values form a *finite quasi-boolean distributive lattice* and the conjunction and disjunction are interpreted as the meet and join operations, respectively, of the lattice. Further, the negation in this logic was De Morgan, ensuring the preservation of involution of negation ($\neg\neg a = a$) and De Morgan laws.

In this paper, we extend χCTL model-checking to reasoning over other logics with non-classical negation, in particular, intuitionistic, minimal and Galois. Such logics are often used for specification and reasoning, and our plan is to develop an automatic verification procedure for cases when the number of truth values is finite. The goals of this paper are: (1) to define χCTL operators directly; (2) to establish relationships between them; and (3) to ensure that these operations can be computed using fixpoint operations. Our results allow us to extend χChek to reasoning over these logics. Further, this work sets the stage for model-checking over various other non-classical logics.

The rest of this paper is organized as follows: Section 2 gives a short introduction to classical CTL model-checking. Section 3 defines the non-classical logics used in this paper. Section 4 formally defines χKripke structures and χCTL operators and studies their properties. The main result of this paper is given in Section 5, where we show that our definitions of χCTL operators satisfy classical fixpoint formulations of their CTL counterparts and discuss adequate sets for χCTL. We conclude in Section 6 with a summary of the paper and directions for future work.

2 Classical CTL Model-Checking

In this section, we give a brief overview of classical CTL model checking. CTL model-checking is an automatic technique for verifying properties expressed in a propositional branching-time temporal logic called *Computation Tree Logic* (CTL) [11].

The language of CTL is the language of propositional logic augmented with unary temporal operators EX, AX, EF, AF, EG, and AG, and binary temporal operators EU and AU:

1. Constants TRUE and FALSE are CTL formulas.
2. Every propositional variable is a CTL formula.
3. If φ and ψ are CTL formulas, then so are $\neg\varphi$, $\varphi\wedge\psi$, $\varphi\vee\psi$, $EX\varphi$, $AX\varphi$, $EF\varphi$, $AF\varphi$, $E[\varphi\ U\ \psi]$, $A[\varphi\ U\ \psi]$, $AG\varphi$, $EG\varphi$.

A model of CTL is a Kripke structure, consisting of a set of states S, a transition relation $R \subseteq S \times S$, an initial state $s_0 \in S$, a set of atomic propositions A, and a labeling function $I : S \rightarrow 2^A$, which maps a state to a set of atomic propositions that are TRUE in it. R must be total, i.e, $\forall s \in S \cdot \exists t \in S \cdot (s,t) \in R$. Properties are evaluated on a tree of infinite computations produced by the model. Finite computations are modeled by adding a self-loop to the final state of the computation. The standard notation $M, s \models \varphi$ indicates that a formula φ is TRUE in a state s of a model M. If a formula is TRUE in the initial state, it is considered to be TRUE in the model.

The logic connectives $\neg$, $\wedge$ and $\vee$ have their usual meanings. The existential and universal quantifiers E and A are used to quantify over paths. The operator X means "in the next state", F represents "sometime in the future", U is "until", and G is "globally". For example, $EX\varphi$ is TRUE in state s if φ is TRUE in some immediate successor of s, while $AX\varphi$ is TRUE if φ is TRUE in every immediate successor of s. $EF\varphi$ is TRUE in s if φ is TRUE in the future along some path from s; $E[\varphi\ U\ \psi]$ is TRUE in s if along some path from s, φ continuously stays TRUE until ψ is TRUE. $EG\varphi$ is TRUE in s if φ is TRUE in every state along some path from s. $AF\varphi$, $A[\varphi\ U\ \psi]$ and $AG\varphi$ are defined similarly, replacing the quantification over some paths by the one over all paths. In summary,

$$
\begin{array}{rcl}
M, s_i \models a & \text{iff} & a \in I(s_i) \\
M, s_i \models \neg\varphi & \text{iff} & M, s_i \not\models \varphi \\
M, s_i \models \varphi \wedge \psi & \text{iff} & M, s_i \models \varphi \ \wedge\ M, s_i \models \psi \\
M, s_i \models \varphi \vee \psi & \text{iff} & M, s_i \models \varphi \ \vee\ M, s_i \models \psi \\
M, s_i \models EX\varphi & \text{iff} & \exists t \in S \cdot (s_i, t) \in R \ \wedge\ M, t \models \varphi \\
M, s_i \models AX\varphi & \text{iff} & \forall t \in S \cdot (s_i, t) \in R \rightarrow M, t \models \varphi \\
M, s_i \models EG\varphi & \text{iff} & \text{there exists some path } s_i, s_{i+1}, \\
 & & \ldots \ \cdot\ \forall j \geq i \cdot M, s_j \models \varphi \\
M, s_i \models AG\varphi & \text{iff} & \text{for every path } s_i, s_{i+1}, \ldots \\
 & & \cdot\ \forall j \geq i \cdot M, s_j \models \varphi \\
M, s_i \models EF\varphi & \text{iff} & \text{there exists some path } s_i, s_{i+1}, \ldots \\
 & & \cdot\ \exists j \geq i \cdot M, s_j \models \varphi \\
M, s_i \models AF\varphi & \text{iff} & \text{for every path } s_i, s_{i+1}, \\
 & & \ldots \ \cdot\ \exists j \geq i \cdot M, s_j \models \varphi \\
M, s_i \models E[\varphi\ U\ \psi] & \text{iff} & \text{there exists some path } s_i, s_{i+1}, \ldots \cdot \\
 & & \exists j \geq i \cdot M, s_j \models \psi \ \wedge \\
 & & \forall k \cdot i \leq k < j \Rightarrow M, s_k \models \varphi \\
M, s_i \models A[\varphi\ U\ \psi] & \text{iff} & for \text{ every path } s_i, s_{i+1}, \ldots \cdot \\
 & & \exists j \geq i \cdot M, s_j \models \psi \ \wedge \\
 & & \forall k \cdot i \leq k < j \Rightarrow M, s_k \models \varphi
\end{array}
$$

The "until" operators are strong, i.e., $E[\varphi\ U\ \psi]$ and $A[\varphi\ U\ \psi]$ are TRUE only if eventually ψ is TRUE.

A number of identities hold between CTL operators [18, 12]. We list a few of them here.

$$
\begin{array}{lcll}
A[\varphi\ U\ \psi] & = & \neg E[\neg\psi\ U\ \neg\varphi \wedge \neg\psi] \ \wedge & \\
 & & \neg EG\neg\psi & (AU \text{ and } EU) \\
AX\varphi & = & \neg EX\neg\varphi & (AX \text{ and } EX) \\
AF\varphi & = & A[\top\ U\ \varphi] & (AF \text{ and } AU) \\
EF\varphi & = & E[\top\ U\ \varphi] & (EF \text{ and } EU) \\
AG\varphi & = & \neg EF\neg\varphi & (AG \text{ and } EF) \\
EG\varphi & = & \neg AF\neg\varphi & (EG \text{ and } AF)
\end{array}
$$

Clearly, many CTL operators are redundant and do not need to be computed directly.

Definition 1 *A set $C = \{c_1, ..., c_i\}$ of temporal CTL operators is* adequate *if the interpretation of all CTL operators can be obtained from the results of the interpretation of elements of C and propositional operators $\{\wedge, \vee, \neg\}$.*

For example, some adequate sets for CTL are $\{EX, EG, EU\}$ and $\{EX, EU, AU\}$ [18, 12]. A complete study of different adequate sets of CTL operators is given in [21, 20].

Alternatively, CTL operators can be described using their fixpoint formulations, as shown in Figure 1. This description is most useful for symbolic model-checking [22]. $\mu Z.f(Z)$ and $\nu Z.f(Z)$ indicate the least and the greatest fixpoints of f, respectively.

3 Logics with Non-Classical Negations

In this section, we discuss the properties of the logics we consider in this paper. Our logics are presumed to have a finite number of truth values. Rather than giving a proof system for each logic, we define conjunction, disjunction and negation on the truth values of the logic. This is appropriate for the goals of model-checking, where we seek to determine properties, expressible in CTL, of particular models. The fact that there is a finite number of truth values guarantees convergence of fixpoints during the computation of CTL formulas over such logics.

We are concerned here with non-classical negation; consequently, classical properties of conjunction and disjunction operators are retained, that is, we presume that conjunction and disjunction are each associative, commutative and idempotent, and that conjunction and disjunction distribute over each other. Such properties are easily guaranteed if we require that the truth values of the logic form a finite distributive lattice, where conjunction and disjunction are meet and join, respectively. In what follows, we use the same symbol for the logical operators and their corresponding lattice operators.

Definition 2 *A* lattice *is a partially-ordered set ($\mathcal{L}$, $\leq$) for which a unique greatest lower bound, denoted by $\wedge$ and referred to as* meet*, and a unique least upper bound, denoted by $\vee$ and referred to as* join*, exist for each finite set of elements. Meet and join over an empty set are referred to as* top *($\top$) and* bottom *($\perp$), respectively.*

We use the equality sign to denote the *identity* relation between different lattice elements:

$$
\begin{array}{rcll}
AG\varphi & = & \nu Z.\varphi \ \wedge \ AXZ & (AG \text{ fixpoint}) \\
EG\varphi & = & \nu Z.\varphi \ \wedge \ EXZ & (EG \text{ fixpoint}) \\
AF\varphi & = & \mu Z.\varphi \vee AXZ & (AF \text{ fixpoint}) \\
EF\varphi & = & \mu Z.\varphi \vee EXZ & (EF \text{ fixpoint}) \\
A[\varphi\ U\ \psi] & = & \mu Z.\psi \vee (\varphi \wedge AXZ) & (AU \text{ fixpoint}) \\
E[\varphi\ U\ \psi] & = & \mu Z.\psi \vee (\varphi \wedge EXZ) & (EU \text{ fixpoint})
\end{array}
$$

Figure 1. Fixpoint formulations of CTL operators.

$$a = b \quad \triangleq \quad a \leq b \ \wedge \ b \leq a \qquad \text{(identity)}$$

Since the join operator gives rise to a partial order (defined as $a \leq b$ iff $a \vee b = b$) we shall henceforth denote a lattice by $(\mathcal{L}, \wedge, \vee)$, and presume that the associated partial order is the one determined by the join. $\wedge$ and $\vee$ are *isotone* operators:

$$
\begin{array}{llll}
\text{if} & a \leq b & \text{then} & a \wedge c \leq b \wedge c \qquad \text{(isotonicity)} \\
\text{if} & a \leq b & \text{then} & c \wedge a \leq c \wedge b \\
\text{if} & a \leq b & \text{then} & a \vee c \leq b \vee c \\
\text{if} & a \leq b & \text{then} & c \vee a \leq c \vee b
\end{array}
$$

In what follows, we often refer to notation $a \leq b$ and $a = b$ to indicate "if a then b" and "a iff b", respectively.

Given a lattice $L = (\mathcal{L}, \wedge, \vee)$, we describe several unary operators with properties of non-classical negation $\neg$. We take our inspiration from the recent work by Dunn [13] who gave the axiomatization of properties of negation for several logics and arranged them to form a partial order, based on the strength of negation. We discuss several logics described in [13]: De Morgan, Galois, minimal and intuitionistic.

Definition 3 *A Galois negation, $\neg = (\neg_r, \neg_l)$, often referred to as a* split *negation, defined over a lattice ($\mathcal{L}$, $\wedge$, $\vee$), is a pair of unary operators, with properties*

$$
\begin{array}{rcll}
a \leq \neg_l\, b & = & b \leq \neg_r\, a & \text{(Galois connection)} \\
\neg_r(a \vee b) & = & \neg_r\, a \wedge \neg_r\, b & \text{(anti-disjunction)} \\
\neg_l(a \vee b) & = & \neg_l\, a \wedge \neg_l\, b & \\
\neg_r\, a \vee \neg_r\, b & \leq & \neg_r(a \wedge b) & \text{(sub-anti-conjunction)} \\
\neg_l\, a \vee \neg_l\, b & \leq & \neg_l(a \wedge b) & \\
a & \leq & \neg_r\, \neg_l\, a & \text{(constructive double negation)} \\
a & \leq & \neg_l\, \neg_r\, a & \\
(a \leq b) & \leq & (\neg_l\, b \leq \neg_l\, a) & \text{(sub-anti-monotonicity)} \\
(a \leq b) & \leq & (\neg_r\, b \leq \neg_r\, a) &
\end{array}
$$

The Galois connection property is equivalent to the four properties in constructive double negation and sub-anti-monotonicity.

Definition 4 *A minimal negation $\neg$, defined over a lattice ($\mathcal{L}$, $\wedge$, $\vee$), is a unary operator with the properties of a Galois negation where $\neg = \neg_r = \neg_l$.*

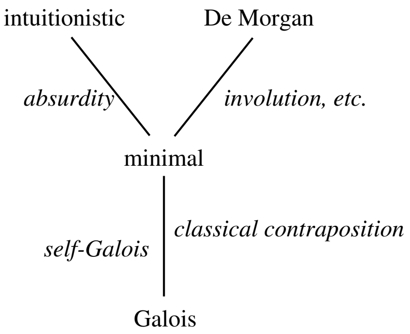


Figure 2. Several logics with non-classical negation.

Definition 5 *An intuitionistic negation $\neg$, defined over a lattice ($\mathcal{L}$, $\wedge$, $\vee$), is a unary operator with the properties of a minimal negation and with the property:*

$$a \wedge \neg a \quad = \quad \bot \qquad \text{(absurdity)}$$

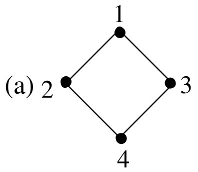
Definition 6 *A De Morgan negation $\neg$ defined over a lattice ($\mathcal{L}$, $\wedge$, $\vee$), is a unary operator with properties*

$$
\begin{array}{rcll}
\neg(a \wedge b) & = & \neg a \vee \neg b & \text{(De Morgan)} \\
\neg(a \vee b) & = & \neg a \wedge \neg b & \\
\neg\neg a & = & a & \text{(involution)} \\
a \leq b & = & \neg b \leq \neg a & \text{(anti-monotonicity)}
\end{array}
$$

Clearly, any property that holds of a Galois negation holds for minimal, intuitionistic and De Morgan negations when we assume $\neg = \neg_r = \neg_l$. In what follows, we often state a property for a Galois negation, where the analogous properties for the logics with stronger negations can be inferred. Further, it can be shown easily that a De Morgan negation has more properties than a minimal negation. However, it does not satisfy the absurdity property of an intuitionistic negation. Figure 2 summarizes the relationship between the different logics we consider here, based on the strength of the negation operator.

For De Morgan, minimal and intuitionistic logics, implication ($\Rightarrow$) and equivalence ($\Leftrightarrow$) are defined using the law of material implication:

$$
\begin{array}{rcll}
a \Rightarrow b & \triangleq & \neg a \vee b & \text{(material implication)} \\
a \Leftrightarrow b & \triangleq & (a \Rightarrow b) \ \wedge \ (b \Rightarrow a) & \text{(equivalence)}
\end{array}
$$



(b)

	$\neg_l$	$\neg_r$
1	4	4
2	3	4
3	4	2
4	1	1

Figure 3. A model of a 4-valued Galois logic: (a) the lattice; (b) the negations table.

For Galois logics, there are two implication (and consequently, two equivalence) operators:

$$
\begin{array}{lcll}
a \Rightarrow_l b & \triangleq & \neg_l\, a \vee b & \text{(material implication)} \\
a \Rightarrow_r b & \triangleq & \neg_r\, a \vee b & \\
a \Leftrightarrow_l b & \triangleq & (a \Rightarrow_l b) \ \wedge\ (b \Rightarrow_l a) & \text{(equivalence)} \\
a \Leftrightarrow_r b & \triangleq & (a \Rightarrow_r b) \ \wedge\ (b \Rightarrow_r a) &
\end{array}
$$

Definition 7 *We say that $L = (\mathcal{L}, \wedge, \vee, \neg)$ is a Galois logic, if it is a multi-valued logic where the truth values form a finite distributive lattice $(\mathcal{L}, \wedge, \vee)$ and $\neg$ denotes a pair of unary operators $(\neg_l, \neg_r)$ that have the properties of the Galois (split) negation.*

We say that $L = (\mathcal{L}, \wedge, \vee, \neg)$ is a minimal, intuitionistic, or De Morgan logic, if it is a multi-valued logic where the truth values form a finite distributive lattice $(\mathcal{L}, \wedge, \vee)$ and the negation is a minimal, intuitionistic, or De Morgan negation, respectively.

For this paper, the term "multi-valued logic" is used to refer to any Galois, minimal, intuitionistic or De Morgan logic. In the literature, De Morgan logics are often referred to as *quasi-boolean* logics [4]. We note that our use of the term "intuitionistic" refers only to logics where the negation has intuitionistic properties. We are not dealing with logics where the implication is intuitionistic (Heyting).

Finally, we establish some properties of tops and bottoms of lattices corresponding to our logics.

Lemma 1 *If $(\neg_r, \neg_l)$ is a Galois negation over a lattice ($\mathcal{L}$, $\wedge$, $\vee$), then*

$$\neg_r \top \ = \ \bot \qquad \neg_l \top \ = \ \bot \qquad \text{(negation of } \top\text{)}$$

Thus, similar properties hold for logic with intuitionistic, minimal and De Morgan negations.

For the remainder of this paper, we assume that the dual property,

$$\neg_r \bot \ = \ \top \qquad \neg_l \bot \ = \ \top \qquad \text{(negation of } \bot\text{)}$$

also holds. Figure 3 gives an example of a 4-valued Galois logic that satisfies the above conditions. In general, negation of $\bot$ always holds for logics with De Morgan negations but does not necessarily hold for logics with weaker negations.

4 Multi-Valued CTL

In this section, we describe multi-valued Kripke structures, which we call χKripke structures [7], give the definition of our multi-valued CTL (χCTL) operators and study their properties.

4.1 χKripke Structures

Definition 8 *$M = (S, s_0, R, I, A, L)$ is a χ*Kripke structure*, where $L = (\mathcal{L}, \wedge, \vee, \neg)$ is a multi-valued logic; A is a (finite) set of propositional variables; S is a (finite) set of states; $s_0 \in S$ is the initial state; $R : S \times S \rightarrow \mathcal{L}$ is a transition relation, assigning degrees of truth to transitions between pairs of states; and $I : S \rightarrow \mathcal{L}^A$ is a (total) interpretation function assigning values to variables in states.*

Note that a χKripke structure is a completely connected weighted graph. We also ensure that there is at least one non-false transition out of each state, extending the classical notion of Kripke structures. Formally,

$$\forall s \in S \cdot \exists t \in S \cdot R(s, t) \neq \bot$$

χCTL is a multi-valued extension of CTL. It has the same syntax as CTL, except that any $l \in \mathcal{L}$ is also a χCTL formula.

We are now ready to give semantics of χCTL operators on a χKripke structure M over a non-classical logic L. We use the same symbol φ to denote both a χCTL formula φ and its interpretation in a χKripke structure. The meaning is clear from the context. Let $\varphi(s)$ denote the value of φ at state s. Finally, if we write $\varphi \leq \psi$, we mean that $\forall s \in S \cdot \varphi(s) \leq \psi(s)$ in any χKripke structure.

For intuitionistic, minimal and Galois logics, we further presume that χCTL has a constant symbol $\top$ and the interpretation in any χKripke structure is subject to the condition that $\forall s \in S \cdot \top(s) = \top$. For minimal and Galois logics, we presume that χCTL has a constant symbol $\bot$ and the interpretation in any χKripke structure is subject to the condition that $\forall s \in S \cdot \bot(s) = \bot$. For intuitionistic logic, for any interpretation and any formula φ, $(\varphi \wedge \neg\varphi)(s) = \bot$.

4.2 Next-Time χCTL Operators

We start by giving semantics of those χCTL operators that do not include fixpoints [9]:

$$
\begin{array}{rcl}
a(s) & \triangleq & (I(s))(a) \\
(\neg\varphi)(s) & \triangleq & \neg(\varphi(s)) \\
(\varphi \wedge \psi)(s) & \triangleq & \varphi(s) \wedge \psi(s) \\
(\varphi \vee \psi)(s) & \triangleq & \varphi(s) \vee \psi(s)
\end{array}
$$

We proceed by defining the EX operator. In classical CTL, this operator is defined using existential quantification over next states. We extend the notion of existential quantification for multi-valued reasoning through the use of disjunction. This treatment of quantification is standard [3, 23]. The semantics of EX is [9]:

$$EX\varphi(s) \triangleq \bigvee_{t\in S}(R(s,t) \wedge \varphi(t)) \qquad \text{(def. of } EX)$$

We define AX using the conjunction in place of the universal quantification.

$$AX\varphi(s) \triangleq \bigwedge_{t\in S}(R(s,t) \Rightarrow \varphi(t)) \qquad \text{(def. of } AX)$$

Note that the operation $R(s,t) \Rightarrow \varphi(t)$ yields $\top$ for those states where $R(s,t)$ is $\bot$. For a Galois logic, we define a pair of AX operators that differ by the type of implication used:

$$\begin{array}{lcll} AX_l\varphi(s) & \triangleq & \bigwedge_{t\in S}(R(s,t) \Rightarrow_l \varphi(t)) & \text{(def. of } AX) \\ AX_r\varphi(s) & \triangleq & \bigwedge_{t\in S}(R(s,t) \Rightarrow_r \varphi(t)) & \end{array}$$

In what follows, we use a common notation for the different AX operators, referring to them as $AX_c \cdot c \in \{r, l, \{\}\}$ to indicate AX_r, AX_l and AX, respectively.

Definition 9 *Let C be a unary operator on χCTL formulas. C is* monotone *if for all χCTL formulas φ and ψ, and for the interpretation in any χKripke structure M,*

$$\textit{if } \forall s \in S \cdot \varphi(s) \leq \psi(s) \textit{ then } \forall s \in S \cdot (C\varphi)(s) \leq (C\psi)(s)$$

The proofs of the next two propositions are omitted due to space restrictions, and appear in [10].

Proposition 1 *The operators EX and AX_c, where $c \in \{r, l, \{\}\}$, are monotone.*

Proposition 2 *Let $c \in \{r, l, \{\}\}$ and either $c_1 = c_2 = \{\}$ or $c_1, c_2 \in \{r, l\} \wedge c_1 \neq c_2$. Then, for a Galois, intuitionistic, De Morgan or minimal logic $(\mathcal{L}, \wedge, \vee, \neg)$, the following are the properties of AX_c and EX:*

$$\begin{array}{rcll} EX\varphi & \leq & \neg_{c_1}\neg_{c_2}EX\varphi & \text{(constructive double negation)} \\ AX_c\varphi & \leq & \neg_{c_1}\neg_{c_2}AX_c\varphi & \\ AX_c\neg_c\varphi & \leq & \neg_c EX\varphi & \text{(sub-}AX\text{ and }EX) \\ AX_{c_1}\varphi & \leq & \neg_{c_1}EX\neg_{c_2}\varphi & \text{(contra }AX\text{ and }EX) \end{array}$$

Proposition 3 *[9] For De Morgan logic, the classical identity between AX and EX holds, that is:*

$$\neg AX\varphi = EX\neg\varphi \qquad (AX \text{ and } EX)$$

4.3 Other χCTL Operators

In this section, we give the semantics for the remaining χCTL operators. The rationale for the definitions presented below is as follows. Classical CTL operators are usually quantified over *reachable* paths in Kripke structures, i.e., $\pi = (s_0, s_1, s_2, ...)$ is considered to be a path if $(s_0, s_1) \in R \wedge (s_1, s_2) \in R \wedge ....$ However, in χKripke structures, the value of $R(s,t)$ ranges over the elements of the lattice, including $\bot$, and we would like to quantify χCTL operators over *all* paths and then factor in the level of reachability. Thus, we replace the classical definitions, e.g., "for all paths $s_0, s_1, s_2, ..., s_i$, $\varphi(s_i)$ holds" and "exists a path $s_0, s_1, s_2, ..., s_i$ such that $\varphi(s_i)$ holds" by the equivalent expressions "for all states $s_0, s_1, s_2, ..., s_i \cdot (R(s_0,s_1) \wedge R(s_1,s_2) \wedge ... \wedge R(s_{i-1},s_i)) \Rightarrow \varphi(s_i)$" and "exist states $s_0, s_1, s_2, ..., s_i$ such that $(R(s_0,s_1) \wedge R(s_1,s_2) \wedge ... \wedge R(s_{i-1},s_i)) \wedge \varphi(s_i)$", respectively.

We start by introducing a formal notion of paths.

Definition 10 *A* path $\pi(s)$ *over a χKripke structure K is a sequence of states $s_0, s_1, s_2, ...$, where $s_0 = s$. Let $\Pi(s)$ be the set of all paths emanating from s.*

Our goal is to use the intuition given in the beginning of this subsection to rewrite definitions of CTL operators to obtain their χCTL counterparts. These definitions can have several forms that are classically equivalent, e.g.

$$\begin{array}{lcl} AF_1\varphi(s_0) & \triangleq & \bigwedge_{\Pi(s_0)}\bigvee_{j\geq 0}(\bigwedge_{i=0}^{j-1}R(s_i,s_{i+1}) \Rightarrow \varphi(s_j)) \\ AF_2\varphi(s_0) & \triangleq & \bigwedge_{\Pi(s_0)}\bigvee_{j\geq 0}(\varphi(s_j) \vee \bigvee_{i=0}^{j-1}\neg R(s_i,s_{i+1})) \end{array}$$

These definitions are also equivalent for logics with De Morgan negations but are different for logics with minimal negation. We now present those formulations that can be computed using fixpoints and study their properties. In Section 5 we prove that in fact they can be computed using the fixpoint formulation.

Definition 11 *Let $c \in \{r, l, \{\}\}$. Then for a Galois, intuitionistic, De Morgan or minimal logic $(\mathcal{L}, \wedge, \vee, \neg)$, the definitions of χCTL operators are as shown in Figure 4.*

Of these definitions, the one for AU is the least intuitive. We have obtained it from the property "AU and EU" in Section 2, using our new definitions of EU and EG. We further propagated negations to the inner-most terms and simplified the resulting expression.

Proposition 4 *Let $c \in \{r, l, \{\}\}$. Then, for a Galois, intuitionistic, De Morgan or minimal logic $(\mathcal{L}, \wedge, \vee, \neg)$, EF and AF_c preserve the classical relationships with EU and AU_c, respectively:*

$$\begin{array}{rcll} AF_c\varphi & = & A_c[\top U \varphi] & (AF \text{ and } AU) \\ EF\varphi & = & E[\top U \varphi] & (EF \text{ and } EU) \end{array}$$

The proof of the above proposition is clear by inspection of the definition of the χCTL operators.

Proposition 5 *Let $c \in \{r, l, \{\}\}$ and either $c_1 = c_2 = \{\}$ or $c_1, c_2 \in \{r, l\} \wedge c_1 \neq c_2$. Then, for a Galois, intuitionistic, De Morgan or minimal logic $(\mathcal{L}, \wedge, \vee, \neg)$, the following are the relationships between the χCTL operators:*

$$\begin{array}{rcll} \varphi & \leq & \neg_{c_1}\neg_{c_2}\varphi & \text{(constructive double negation)} \\ AF_c\neg_c\varphi & \leq & \neg_c EG\varphi & \text{(sub-}AF\text{ and }EG) \\ AF_{c_1}\varphi & \leq & \neg_{c_1}EG\neg_{c_2}\varphi & \text{(contra }AF\text{ and }EG) \\ AG_c\neg_c\varphi & \leq & \neg_c EF\varphi & \text{(sub-}AG\text{ and }EF) \\ AG_{c_1}\varphi & \leq & \neg_{c_1}EF\neg_{c_2}\varphi & \text{(contra }AG\text{ and }EF) \end{array}$$

$$
\begin{array}{rcll}
EF\varphi(s_0) & \triangleq & \bigvee_{\Pi(s_0)} \bigvee_{j\geq 0}(\varphi(s_j) \wedge \bigwedge_{i=0}^{j-1} R(s_i, s_{i+1})) & \text{(def. of } EF\text{)} \\
AF_c\varphi(s_0) & \triangleq & \bigwedge_{\Pi(s_0)} \bigvee_{j\geq 0}(\varphi(s_j) \vee \bigvee_{i=0}^{j-1} \neg_c R(s_i, s_{i+1})) & \text{(def. of } AF\text{)} \\
EG\varphi(s_0) & \triangleq & \bigvee_{\Pi(s_0)} \bigwedge_{j\geq 0}(\varphi(s_j) \wedge \bigwedge_{i=0}^{j-1} R(s_i, s_{i+1})) & \text{(def. of } EG\text{)} \\
AG_c\varphi(s_0) & \triangleq & \bigwedge_{\Pi(s_0)} \bigwedge_{j\geq 0}(\varphi(s_j) \vee \bigvee_{i=0}^{j-1} \neg_c R(s_i, s_{i+1})) & \text{(def. of } AG\text{)} \\
E[\varphi U \psi](s_0) & \triangleq & \bigvee_{\Pi(s_0)} \bigvee_{j\geq 0}(\bigwedge_{i=0}^{j-1} R(s_i, s_{i+1}) \wedge \psi(s_j) \wedge \bigwedge_{i=0}^{j-1} \varphi(s_i)) & \text{(def. of } EU\text{)} \\
A_c[\varphi U \psi](s_0) & \triangleq & \bigwedge_{\Pi(s_0)} \bigwedge_{j\geq 0}((\bigvee_{i=0}^{j-1} \neg_c R(s_i, s_{i+1})) \vee \psi(s_j) \wedge \bigwedge_{i=0}^{j} \varphi(s_i))) & \text{(def. of } AU\text{)}
\end{array}
$$

Figure 4. Definitions of χCTL operators for Galois, intuitionistic, De Morgan or minimal logics.

The proof of the above proposition is similar to the proof of Proposition 2 and is again omitted.

Proposition 6 *[9] For De Morgan logic, the following additional properties hold:*

$$
\begin{array}{rcll}
EF\varphi & = & \neg AG\neg\varphi & (AG \text{ and } EF) \\
EG\varphi & = & \neg AF\neg\varphi & (EG \text{ and } AF) \\
A[\varphi U \psi] & = & \neg E[\neg\psi\ U\ \neg\varphi \wedge \neg\psi] \wedge & \\
 & & \neg EG\neg\psi & (AU \text{ and } EU)
\end{array}
$$

5 χCTL Model-Checking

Symbolic model-checking over non-classical logics is possible only if *all* operators can be computed using their fixpoint formulations. Otherwise, only a subset of χCTL is analyzable. In this section, we prove that the definitions of χCTL operators given in Definition 11 are equivalent to their fixpoint formulations and look at adequate sets for χCTL.

Theorem 1 *Let $c \in \{r, l, \{\}\}$ and let χCTL operators $\{EF, AF_c, EG, AG_c, EU, AU_c\}$ be given by Definition 11. Then, for a Galois, intuitionistic, De Morgan or minimal logic $(\mathcal{L}, \wedge, \vee, \neg)$, they are equivalent to their fixpoint formulations, i.e.*

$$
\begin{array}{rcll}
AG_c\varphi & = & \nu Z.\varphi\ \wedge\ AX_cZ & (AG \text{ fixpoint}) \\
EG\varphi & = & \nu Z.\varphi\ \wedge\ EXZ & (EG \text{ fixpoint}) \\
AF_c\varphi & = & \mu Z.\varphi \vee AX_cZ & (AF \text{ fixpoint}) \\
EF\varphi & = & \mu Z.\varphi \vee EXZ & (EF \text{ fixpoint}) \\
A_c[\varphi\ U\ \psi] & = & \mu Z.\psi \vee (\varphi \wedge AX_cZ) & (AU \text{ fixpoint}) \\
E[\varphi\ U\ \psi] & = & \mu Z.\psi \vee (\varphi \wedge EXZ) & (EU \text{ fixpoint})
\end{array}
$$

The proof of this theorem is omitted due to space limitations. The proof for AF_c is available in the extended version of this paper [10]. Note that the theorem holds even if $\mathcal{L}$ is an *arbitrary* complete distributive lattice.

The termination of the algorithm for $AF_c\varphi$ is guaranteed by the usual application of the Knaster-Tarski theorem since our logics are defined over finite lattices.

Proposition 7 *Let $c \in \{r, l, \{\}\}$ and let χCTL operators be given by Definition 11. Then, $\{EU, AU_c, EG, AG_c, EX, AX_c\}$ is an adequate set for χCTL.*

The proof depends on Proposition 4 which indicates that AF_c and EF can be computed from AU_c and EU, respectively.

Proposition 8 *For logics with De Morgan negation, an adequate set for χCTL coincides with one for CTL: $\{EG, EU, EX\}$.*

This result depends on Propositions 3 and 6 and was proved in [7].

In earlier work, Chechik et al. built a symbolic multi-valued model-checker χChek [8] which analyzes χCTL defined over logics with De Morgan negation. The model-checker uses decision diagrams to represent sets of states of a χKripke structure and perform fixpoint computations. It also uses $\{EG, EU, EX\}$ as the adequate set.

The verification engine of χChek can be extended to handle a larger adequate set, in particular, the one that consists of χCTL operators $\{EF, AF_c, EG, AG_c, EU, AU_c\}$. The fixpoint algorithms for computing these operators directly are given in Theorem 1. Work is required to extend χChek's counter-example capabilities[1] to this new adequate set, but we believe this does not present a major challenge.

In the remainder of this section, we look at fragments of χCTL for which adequate sets are smaller than the one given by Proposition 7. We consider two examples, each using a logic with minimal negation.

Example 1. Suppose we know that $(EGEF\varphi)(s) = \top$, where φ is some χCTL formula. Then,

$$
\begin{array}{cl}
 & (\neg EGEF\varphi)(s) \\
\geq & (\text{sub-}AF \text{ and } EG) \\
 & (AF\neg EF\varphi)(s) \\
\geq & (\text{sub-}AG \text{ and } EF) \\
 & (AFAG\neg\varphi)(s)
\end{array}
$$

Therefore,

$$
\begin{array}{cl}
 & (\neg AFAG\neg\varphi)(s) \\
\geq & (\text{sub-anti-monotonicity}) \\
 & (\neg\neg EGEF\varphi)(s) \\
\geq & (\text{constructive double negation}) \\
 & (EGEF\varphi)(s) \\
= & (\text{original assumption}) \\
 & \top
\end{array}
$$

[1]A counter-example is a sequence of states starting from the initial one that exhibits the failure of a property.

if	$(C_1...C_n\varphi)(s)$	=	$\top$	then	$\neg(f(C_1)...f(C_n)\neg\varphi)(s)$	=	$\top$	and	$(f(C_1)...f(C_n)\neg\varphi)(s)$	=	$\bot$
if	$(C_1...C_n\neg\varphi)(s)$	=	$\top$	then	$\neg(f(C_1)...f(C_n)\varphi)(s)$	=	$\top$	and	$(f(C_1)...f(C_n)\varphi)(s)$	=	$\bot$
if	$(O_1...O_n\neg\varphi)(s)$	=	$\top$	then	$\neg(g(O_1)...g(O_n)\varphi)(s)$	=	$\top$	and	$(g(O_1)...g(O_n)\varphi)(s)$	=	$\bot$
if	$(O_1...O_n\varphi)(s)$	=	$\top$	then	$\neg(g(O_1)...g(O_n)\neg\varphi)(s)$	=	$\top$	and	$(g(O_1)...g(O_n)\neg\varphi)(s)$	=	$\bot$

Figure 5. Restricted adequacy conditions for Theorem 2.

Thus, $(\neg AFAG\neg\varphi)(s) = \top$.

Example 2. Suppose we know that $(AFAG\neg\varphi)(s) = \top$, where φ is some χCTL formula. Then,

$$\begin{array}{ll} & (AFAG\neg\varphi)(s) \\ \leq & \text{(sub-}AG\text{ and }EF\text{)} \\ & (AF\neg EF\varphi)(s) \\ \leq & \text{(sub-}AF\text{ and }EG\text{)} \\ & (\neg EGEF\varphi)(s) \end{array}$$

Thus, $(\neg EGEF\varphi)(s) = \top$ and $(EGEF\varphi)(s) = \bot$.

These results allow us to formulate the following theorem, the proof of which is based on Proposition 5 and the monotonicity of χCTL operators. Suppose we have a function f over χCTL operators that converts AF, AG, AX to EG, EF and EX, respectively, and let g be its inverse.

Theorem 2 *(Restricted adequacy) Let L be a minimal logic. Let φ be a CTL formula and let $C_i \in \{EG, EF, EX\}$ and $O_i \in \{AG, AF, AX\}$. Then the conditions given in Figure 5 hold.*

A similar theorem holds for a logic with Galois negation.

Clearly, more work needs to be done to identify further syntactic restrictions on χCTL formulas that allow us to reduce the size of the adequate set of χCTL. In particular, we have not looked at the relationships between the EU and AU operators. We plan to address these shortcomings in future work.

Note that the definition of the χCTL operator that we selected was based on the goal of having an equivalent fixpoint formulation for it. Consider the definitions of AF_1 and AF_2 given in Section 4. AF_2 coincides with the definition we chose for χCTL. In De Morgan negation, $AF_1\varphi$ has the fixpoint property – that is, $AF_1\varphi = \nu Z.\varphi \vee (AXZ)$ (negations can be pushed in the same way as in classical negation). If we weaken the negation to intuitionistic or minimal negation, we can show only that $AF_1(\varphi)(s) \leq (\varphi \wedge AXAF_1\varphi)(s)$. So, while we can compute $(\mu Z.\varphi \vee AXZ)(s)$, we are not computing $AF_1\varphi(s)$.

Thus, we need to formulate our χCTL definitions carefully to get the fixpoint properties we want, so that the answers given by symbolic model-checking correspond to the meaning of χCTL operators.

6 Summary and Future Work

In this paper, we studied CTL model-checking, as defined over multi-valued logics with non-classical negation (χCTL). In particular, we looked at finite-valued logics with De Morgan, intuitionistic, minimal and Galois negations. We identified the cases where classical relationships between CTL operators are preserved in the multi-valued case and where these relationships break down. We also gave definitions for χCTL operators under which they can be computed directly using fixpoint operations and showed how to modify the existing multi-valued model-checker χChek to reason about this version of χCTL. Finally, we discussed the adequate sets for our χCTL as well as alternative definitions of χCTL operators.

In future work, we would like to continue studying the relationship between different definitions of χCTL operators and adequate sets for symbolic computation of subsets of χCTL. In particular, we would like to study logics with an *infinite* number of values for which fixpoint operations converge in finite time.

We are also interested in extending our work to a hierarchy of substructural logics [25]. The substructural hierarchy is built around the idea of weakening properties of the conjunction operator. We may consider premises as resources to be used or "consumed"; consequently, these logics are often referred to as "resource logics", and their applicability to various settings becomes clear. Putting these properties in an algebraic (rather than proof-theoretic) form, we say that to account for the number of times an assumption is used in a proof we may do away with idempotence of $\wedge$; specifically, we no longer assume that $a \leq a \wedge a$; to account for the order in which assumptions are used in a proof, we no longer assume that commutativity holds ($a \wedge b = b \wedge a$); to account for whether or not an assumption is used in a proof, we no longer assume the lower bound property ($a \wedge b \leq a$); to account for the grouping of resources, the associativity of $\wedge$ is dropped.

In this paper, we assumed that implication is material implication but intuitionistic implication, expressed by the rule:

$$b \leq a \Rightarrow c \quad \text{iff} \quad a \wedge b \leq c \qquad \text{(residuation)}$$

has semantics of interest in specifications. We are interested in studying properties of χCTL defined over logics with different implications.

We note that Huth and Pradhan [17] have defined complete AC lattices as structures $(\mathcal{L}_a, \leq_a, \neg_a, \mathcal{L}_c, \leq_c, \neg_c)$, where $\leq_a$ and $\leq_c$ are partial orders over their respective domains, inducing lattice structures and satisfying split classical double negation properties ($\neg_a \neg_c \varphi = \varphi$ and $\neg_c \neg_a \varphi = \varphi$) and sub-anti-monotonicity properties ($\varphi \leq_a \psi$ implies $\neg_a \psi \leq_c \neg_a \varphi$ and $\varphi \leq_c \psi$ implies $\neg_c \psi \leq_c \neg_a \varphi$). They showed that three-valued model-checking and AC-lattices as their denotation spaces can be systematically lifted to property verification and requirements elicitation for multiple stakeholders. We believe that the logics considered in our paper may have the same application.

Acknowledgments

The example in Figure 3 was generated by Albert Lai using Daniel Jackson's tool Alloy [19]. Arie Gurfinkel helped prove Theorem 1. We also thank anonymous referees as well as Arie and Albert, who made many excellent suggestions to help improve this paper. We gratefully acknowledge the financial support provided by NSERC and CITO. This work was done while Wendy MacCaull was on sabbatical at the University of Toronto.

References

[1] M. Abadi and G. Plotkin. "A Logical View of Composition and Refinement". In *Proceedings of the 18th Annual Symposium on Principles of Programming Languages*, pages 323–332, 1991.

[2] S. Abramsky. "Computational Interpretations of Linear Logic". *Theoretical Computer Science*, 111(1-2):3–57, April 1993.

[3] N. Belnap. "A Useful Four-Valued Logic". In Dunn and Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 30–56. Reidel, 1977.

[4] L. Bolc and P. Borowik. *Many-Valued Logics*. Springer-Verlag, 1992.

[5] G. Bruns and P. Godefroid. "Model Checking Partial State Spaces with 3-Valued Temporal Logics". In *Proceedings of 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.

[6] M. Chechik, B. Devereux, and S. Easterbrook. "Implementing a Multi-Valued Symbolic Model-Checker". In *Proceedings of 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'01)*, volume 2031 of *Lecture Notes in Computer Science*, pages 404–419. Springer, April 2001.

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

[8] 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)*, Lecture Notes in Computer Science, pages 505–509, Copenhagen, Denmark, July 2002. Springer.

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

[10] M. Chechik and W. MacCaull. "On Model-Checking over Logics with Non-Classical Negations". CSRG Tech Report 449, University of Toronto, October 2002.

[11] E. Clarke, E. Emerson, and A. 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.

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

[13] J. Dunn. "A Comparative Study of Various Model-Theoretic Treatments of Negation: A History of Formal Negation". In D. Gabbay and H. Wansing, editors, *What is Negation*. Kluwer Academic Publishers, 1999.

[14] S. Easterbrook and M. Chechik. "A Framework for Multi-Valued Reasoning over Inconsistent Viewpoints". In *Proceedings of International Conference on Software Engineering (ICSE'01)*, pages 411–420, Toronto, Canada, May 2001. IEEE Computer Society Press.

[15] A. Gurfinkel, B. Devereux, and M. Chechik. "Model Exploration with Temporal Logic Query Checking". In *Proceedings of SIGSOFT Conference on Foundations of Software Engineering (FSE'02)*, pages 139–148, Charleston, South Carolina, November 2002. ACM Press.

[16] 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.

[17] M. Huth and S. Pradhan. "Lifting Assertion and Consistency Checkers from Single to Multiple Viewpoints". Submitted for publication, January 2002.

[18] M. Huth and M. Ryan. *Logic in Computer Science: Modeling and Reasoning About Systems*. Cambridge University Press, 2000.

[19] D. Jackson. "Automating First-Order Relational Logic". In *Proceedings of ACM SIGSOFT Conference on Foundations of Software Engineering (FSE'00)*, San Diego, CA, USA, November 2000.

[20] F. Laroussinie. "About the Expressive Power of CTL Combinators". *Information Processing Letters*, 54:343–345, 1995.

[21] A. Martin. "Adequate Sets of Temporal Connectives in CTL". In *Proceedings of 8th International Workshop on Expressiveness in Concurrency (EXPRESS'01)*, August 2001.

[22] K. McMillan. *Symbolic Model Checking*. Kluwer Academic, 1993.

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

[24] M. Sagiv, T. Reps, and R. Wilhelm. "Parametric Shape Analysis via 3-Valued Logic". In *Proceedings of 26th Annual ACM Symposium on Principles of Programming Languages*, pages 105–118, New York, NY, 1999. ACM.

[25] P. Schroeder-Heister and K. Došen, editors. *Substructural Logics*. Oxford Science Publications, 1993.