Soundness and Completeness of a Model-Checking Proof System for CTL

Georg Friedrich Schuppe Dilian Gurov
KTH Royal Institute of Technology, Stockholm
{schuppe,dilian}@kth.se

September 11, 2023

Abstract

We propose a local model-checking proof system for a fragment of CTL. The rules of the proof system are motivated by the well-known fixed-point characterisation of CTL based on unfolding of the temporal operators. To guarantee termination of proofs, we tag the sequents of our proof system with the set of states that have already been explored for the respective temporal formula. We define the semantics of tagged sequents, and then state and prove soundness and completeness of the proof system, as well as termination of proof search for finite-state models.

1 Introduction

Computation Tree Logic (CTL) is a well-known branching-time temporal logic [3, 5]. Many useful temporal specification patterns can be expressed naturally in CTL. The logic is supported by numerous off-the-shelf model checking tools such as nuSMV [2].

The standard, *global* approach to model checking of a CTL formula ϕ w.r.t. a given state s of a given Kripke structure $\mathcal{M}$ is to first compute the set $[\![\phi]\!]^{\mathcal{M}}$ of all states that satisfy the formula, i.e., the *denotation* of ϕ, and then to check whether $s \in [\![\phi]\!]^{\mathcal{M}}$. This approach allows the use of *symbolic* representations of the denotations of the formula and its subformulas, typically as BDDs (as in nuSMV).

An alternative, *local* approach is to start with the state s and incrementally explore its neighbourhood as required by the formula ϕ, by *unfolding* the latter step-by-step. One obvious advantage of this approach is that it only explores the part of the model that is required to establish or reject the checked formula. Another advantage is that local model checking can be phrased as proof search in a *deductive proof system*. It can then be implemented in a straightforward manner in a logic programming environment such as Prolog. This can be very useful for *education purposes*, since it gives the opportunity for students to create, without much effort, an own tool that can analyse non-trivial models

of system behaviour (typically with up to a few thousand states). In fact, the model-checking proof system presented here has been developed for and used in the course *Logic for Computer Scientists*, given at KTH Royal Institute of Technology, Stockholm.

It is well-known that CTL can be embedded into the (alternation-free fragment of the) modal μ-calculus [6]. Since local model checking proof systems have already been proposed for the latter logic, as for instance in [1], designing one for CTL based on the embedding should be straightforward. However, there are good reasons for designing a self-standing proof system, like the one we propose here. The foremost reason for us has been to utilise the circumstance that it is the alternation-free fragment of the modal μ-calculus that we need to take into account. This suggests that the approach to guaranteeing termination of proof search employed in [6] of *tagging* formulas with the set of states that have already been explored w.r.t. the formula (in this case essentially requiring only the outermost fixed-point needs to be tagged) can be lifted from the level of formulas to the level of sequents. Thus, tagged sequents need to be given a formal semantics, to allow to state formally soundness and completeness of the proof system, and to argue for termination of proof search for finite-state models.

Since our proof system has originally been designed for education purposes, to keep the presentation simple, we have chosen not to include the Until operator of CTL in our treatment, and leave its addition as an exercise to the interested reader. This does not present any technical difficulties, and simply follows the pattern of the other temporal operators and their fixed-point characterisation.

2 Syntax and Semantics of the Logic

We start by presenting the syntax and semantics of the logic, which we call CTL^-, since it is a fragment of CTL.

Definition 2.1 (Logic Syntax)**.** The language is defined over a set of atomic propositions *Atoms*, ranged over by p, as follows:

$$\begin{array}{l} \phi \ ::= \ p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid A\psi \mid E\psi \\ \psi \ ::= \ X\phi \mid G\phi \mid F\phi \end{array}$$

The formulas ϕ are called *state formulas* and ψ *path formulas*. The strict alternation of path and state quantifiers gives rise to six combinations. Notice that negation is only allowed over atomic propositions. The reason for this is that it is cumbersome to come up with a rule for negated formulas in Section 3. However, this restriction does not affect the expressiveness of the logic, since negated formulas can be "deMorganised" so as to push the negation to the atomic propositions.

Definition 2.2 (Kripke Structure)**.** A *Kripke structure* is a tuple $\mathcal{M} = (S, \rightarrow, L)$, where S is a set of *states*, $\rightarrow$ a binary *transition relation* on S, and L :

$S \to 2^{Atoms}$ a *labelling function* that assigns to every state the set of atomic propositions that are deemed true in that state.

Given a Kripke structure, the semantics of a CTL formula ϕ is defined as the set $[\![\phi]\!]^{\mathcal{M}} \subseteq S$ of states that satisfy the formula, sometimes referred to as its *denotation*. Inspired by [1], however, we shall define this notion relative to a set $U \subseteq S$ of states, called a *tag*. We will use such tags in Section 3 to guarantee finiteness of proof trees. Only formulas starting with a temporal operator will need (non-empty) tags.

Definition 2.3 (Logic Semantics)**.** Let $\mathcal{M} = (S, \to, L)$ be a Kripke structure. The semantics of formulas is inductively defined by the following equations:

$$[\![p]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} \{s \in S \mid p \in L(s)\} \tag{1}$$

$$[\![\neg p]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} S \setminus [\![p]\!]^{\mathcal{M}}_{\varnothing} \tag{2}$$

$$[\![\phi \wedge \psi]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} [\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cap [\![\psi]\!]^{\mathcal{M}}_{\varnothing} \tag{3}$$

$$[\![\phi \vee \psi]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} [\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cup [\![\psi]\!]^{\mathcal{M}}_{\varnothing} \tag{4}$$

$$[\![EX\phi]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} pre_{\exists}([\![\phi]\!]^{\mathcal{M}}_{\varnothing}) \tag{5}$$

$$[\![AX\phi]\!]^{\mathcal{M}}_{\varnothing} \stackrel{\text{def}}{=} pre_{\forall}([\![\phi]\!]^{\mathcal{M}}_{\varnothing}) \tag{6}$$

$$[\![EF\phi]\!]^{\mathcal{M}}_{U} \stackrel{\text{def}}{=} \mu Y.([\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cup pre_{\exists}(Y) \setminus U) \tag{7}$$

$$[\![AF\phi]\!]^{\mathcal{M}}_{U} \stackrel{\text{def}}{=} \mu Y.([\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cup pre_{\forall}(Y) \setminus U) \tag{8}$$

$$[\![EG\phi]\!]^{\mathcal{M}}_{U} \stackrel{\text{def}}{=} \nu Y.([\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cap pre_{\exists}(Y) \cup U) \tag{9}$$

$$[\![AG\phi]\!]^{\mathcal{M}}_{U} \stackrel{\text{def}}{=} \nu Y.([\![\phi]\!]^{\mathcal{M}}_{\varnothing} \cap pre_{\forall}(Y) \cup U) \tag{10}$$

where the state transformers $pre_{\exists} : S \to S$ and $pre_{\forall} : S \to S$, and the *least* and *greatest fixed-point* $\mu Y.f(Y)$ and $\nu Y.f(Y)$ of a monotone function $f : S \to S$ are defined as follows:

$$pre_{\exists}(Y) \stackrel{\text{def}}{=} \{s \in S \mid \exists s' \in Y.\ s \to s'\} \tag{11}$$

$$pre_{\forall}(Y) \stackrel{\text{def}}{=} \{s \in S \mid \forall s' \in S.\ (s \to s' \Rightarrow s' \in Y)\} \tag{12}$$

$$\mu Y.f(Y) \stackrel{\text{def}}{=} \bigcap \{X \subseteq S \mid f(X) \subseteq X\} \tag{13}$$

$$\nu Y.f(Y) \stackrel{\text{def}}{=} \bigcup \{X \subseteq S \mid f(X) \supseteq X\} \tag{14}$$

If the tag U is empty, the semantics coincides with the standard semantics of CTL. The semantic rules for $[\![EF\phi]\!]^{\mathcal{M}}_{\varnothing}$, $[\![AF\phi]\!]^{\mathcal{M}}_{\varnothing}$, $[\![EG\phi]\!]^{\mathcal{M}}_{\varnothing}$ and $[\![AG\phi]\!]^{\mathcal{M}}_{\varnothing}$ fall back on known embeddings of CTL into the modal μ-calculus [4].

We shall later need the following result.

Lemma 2.1 (Reduction Lemma [1])**.** For any monotone function ψ on a powerset $Pow(D)$, and any $p \in D$, we have:

$$p \in \mu Y.\psi(Y) \;\Leftrightarrow\; p \in \psi(\mu Y.(\psi(Y) \setminus \{p\})) \tag{15}$$

$$p \in \nu Y.\psi(Y) \;\Leftrightarrow\; p \in \psi(\nu Y.(\psi(Y) \cup \{p\})) \tag{16}$$

The right-hand sides of these logical equivalences involve a slightly modified unfolding of the fixed points: For the least fixed point of a single element, p is removed in the unfolding; for the greatest it is added.

3 A Local Model-Checking Proof System

$$p\ \frac{-}{\mathcal{M}, s \vdash_\varnothing p}\ p \in L(s) \qquad \neg p\ \frac{-}{\mathcal{M}, s \vdash_\varnothing \neg p}\ p \notin L(s)$$

$$\wedge\ \frac{\mathcal{M}, s \vdash_\varnothing \phi \qquad \mathcal{M}, s \vdash_\varnothing \psi}{\mathcal{M}, s \vdash_\varnothing \phi \wedge \psi}$$

$$\vee_1\ \frac{\mathcal{M}, s \vdash_\varnothing \phi}{\mathcal{M}, s \vdash_\varnothing \phi \vee \psi} \qquad \vee_2\ \frac{\mathcal{M}, s \vdash_\varnothing \psi}{\mathcal{M}, s \vdash_\varnothing \phi \vee \psi}$$

$$\mathsf{AX}\ \frac{\mathcal{M}, s_1 \vdash_\varnothing \phi \quad \cdots \quad \mathcal{M}, s_n \vdash_\varnothing \phi}{\mathcal{M}, s \vdash_\varnothing \mathsf{AX}\,\phi}$$

$$\mathsf{AG}_1\ \frac{-}{\mathcal{M}, s \vdash_U \mathsf{AG}\,\phi}\ s \in U \qquad \mathsf{AF}_1\ \frac{\mathcal{M}, s \vdash_\varnothing \phi}{\mathcal{M}, s \vdash_U \mathsf{AF}\,\phi}\ s \notin U$$

$$\mathsf{AG}_2\ \frac{\mathcal{M}, s \vdash_\varnothing \phi \qquad \mathcal{M}, s_1 \vdash_{U,s} \mathsf{AG}\,\phi \quad \cdots \quad \mathcal{M}, s_n \vdash_{U,s} \mathsf{AG}\,\phi}{\mathcal{M}, s \vdash_U \mathsf{AG}\,\phi}\ s \notin U$$

$$\mathsf{AF}_2\ \frac{\mathcal{M}, s_1 \vdash_{U,s} \mathsf{AF}\,\phi \quad \cdots \quad \mathcal{M}, s_n \vdash_{U,s} \mathsf{AF}\,\phi}{\mathcal{M}, s \vdash_U \mathsf{AF}\,\phi}\ s \notin U$$

$$\mathsf{EX}\ \frac{\mathcal{M}, s' \vdash_\varnothing \phi}{\mathcal{M}, s \vdash_\varnothing \mathsf{EX}\,\phi} \qquad \mathsf{EG}_1\ \frac{-}{\mathcal{M}, s \vdash_U \mathsf{EG}\,\phi}\ s \in U$$

$$\mathsf{EG}_2\ \frac{\mathcal{M}, s \vdash_\varnothing \phi \qquad \mathcal{M}, s' \vdash_{U,s} \mathsf{EG}\,\phi}{\mathcal{M}, s \vdash_U \mathsf{EG}\,\phi}\ s \notin U$$

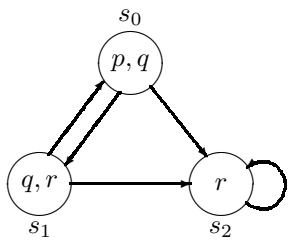
$$\mathsf{EF}_1\ \frac{\mathcal{M}, s \vdash_\varnothing \phi}{\mathcal{M}, s \vdash_U \mathsf{EF}\,\phi}\ s \notin U \qquad \mathsf{EF}_2\ \frac{\mathcal{M}, s' \vdash_{U,s} \mathsf{EF}\,\phi}{\mathcal{M}, s \vdash_U \mathsf{EF}\,\phi}\ s \notin U$$

Figure 1: A Local Model-Checking Proof System for CTL$^-$.

We present our model-checking procedure in the form of a deductive system, consisting of rules over sequents $\mathcal{M}, s \vdash_U \phi$. To guarantee finiteness of proof

trees, and with this completeness of the proof systems as well as termination of proof search, we equip our sequents with *tags* $U \subseteq S$ as already introduced in Section 2. The rules of our proof system are presented in Figure 1. In the premises of the A-rules, $s_1, \ldots, s_n$ denote *all* successors of state s in the Kripke structure $\mathcal{M}$, while in the premises of the E-rules, s' denotes *some* successor of s. To prove that a state s in a Kripke structure $\mathcal{M}$ satisfies a formula ϕ of the logic, one needs to derive the sequent $\mathcal{M}, s \vdash_{\varnothing} \phi$, where the tag is initially empty.

Example. Consider the following Kripke structure:



We would like to show that the formula $\mathsf{EF}\ (\mathsf{EG}\ r)$ holds in state s_0 of the Kripke structure. This can be established by the following proof tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{-}{\mathcal{M}, s_2 \vdash_{\varnothing} r}\ p \quad \dfrac{-}{\mathcal{M}, s_2 \vdash_{[s_2]} \mathsf{EG}\ r}\ \mathsf{EG}_1}{\mathcal{M}, s_2 \vdash_{\varnothing} \mathsf{EG}\ r}\ \mathsf{EG}_2}{\mathcal{M}, s_2 \vdash_{[s_0]} \mathsf{EF}\ (\mathsf{EG}\ r)}\ \mathsf{EF}_1}{\mathcal{M}, s_0 \vdash_{\varnothing} \mathsf{EF}\ (\mathsf{EG}\ r)}\ \mathsf{EF}_2$$

To define and prove the properties of the proof system we need the following semantic notion.

Definition 3.1 (Sequent Validity)**.** A sequent $\mathcal{M}, s \vdash_U \phi$ is termed *valid*, denoted $\mathcal{M}, s \models_U \phi$, iff $s \in [\![\phi]\!]^{\mathcal{M}}_U$.

4 Soundness of the Proof System

A deductive system is termed *sound* if all sequents derivable with its rules are semantically valid (that is, only valid sequents can be proved). We will show that all rules of our proof system preserve the validity of sequents whenever the respective side condition holds. For readability, we omit the superscript $\mathcal{M}$, when it is clear which $\mathcal{M}$ is meant. We abbreviate $U \cup \{s\}$ by writing U, s.

Theorem 4.1 (Soundness)**.** The proof system from Figure 1 is sound.

Proof. The result is a direct consequence of the soundness of each rule, which we show here. Recall that a rule is termed sound if its conclusion is a valid

sequent whenever all its premises are valid and the side-conditions hold.

Rule p. If $p \in L(s)$, the conclusion $\mathcal{M}, s \models_\varnothing p$ follows directly from Definition 3.1 and (1). The argument for rule $\neg p$ is dual.

Rule $\wedge$. We show that $\mathcal{M}, s \models_\varnothing \phi$ and $\mathcal{M}, s \models_\varnothing \psi$ imply $\mathcal{M}, s \models_\varnothing \phi \wedge \psi$. If $s \in [\![\phi]\!]_\varnothing$ and $s \in [\![\psi]\!]_\varnothing$, then obviously

$$s \in [\![\phi]\!]_\varnothing \cap [\![\psi]\!]_\varnothing \overset{(3)}{=} [\![\phi \wedge \psi]\!]_\varnothing.$$

Rules $\vee_1, \vee_2$. We show that $\mathcal{M}, s \models_\varnothing \phi$ implies $\mathcal{M}, s \models_\varnothing \phi \vee \psi$. If $s \in [\![\phi]\!]_\varnothing$, then

$$s \in [\![\phi]\!]_\varnothing \cup [\![\psi]\!]_\varnothing \overset{(4)}{=} [\![\phi \vee \psi]\!]_\varnothing.$$

The argument for $\vee_2$ is similar.

Rule EX. We show that $\mathcal{M}, s' \models_\varnothing \phi$ implies $\mathcal{M}, s \models_\varnothing EX\phi$, where $s \to s'$ with $s, s' \in S$. If $s' \in [\![\phi]\!]_\varnothing$, then with (11), it holds that

$$s \in pre_\exists([\![\phi]\!]_\varnothing) \overset{(5)}{=} [\![EX\phi]\!]_\varnothing.$$

The reasoning for AX and $pre_\forall$ is similar.

Rule EG_1. If $s \in U$, then $s \in [\![EG\phi]\!]_U$ holds by (9).

Rule EG_2. We show that $\mathcal{M}, s \models_\varnothing \phi$ and $\mathcal{M}, s' \models_{U,s} EG\phi$ imply $\mathcal{M}, s \models_U EG\phi$ when $s \notin U$, where $s \to s'$ with $s, s' \in S$. Applying the appropriate semantic rule, and unfolding the fixed point once using Lemma 2.1, we get the equivalences

$$\begin{aligned}
& s \in [\![EG\phi]\!]_U \\
\overset{(9)}{\Leftrightarrow}\ & s \in \nu Y.([\![\phi]\!]_\varnothing \cap pre_\exists(Y) \cup U) \\
\overset{(16)}{\Leftrightarrow}\ & s \in [\![\phi]\!]_\varnothing \cap pre_\exists(\nu Y.([\![\phi]\!]_\varnothing \cap pre_\exists(Y) \cup U \cup \{s\})) \cup U \\
\overset{(9)}{\Leftrightarrow}\ & s \in [\![\phi]\!]_\varnothing \cap pre_\exists([\![EG\phi]\!]_{U,s}) \cup U
\end{aligned}$$

Since we assume $\mathcal{M}, s' \models_{U,s} EG\phi$, we have $s' \in [\![EG\phi]\!]_{U,s}$ and thus $s \in pre_\exists([\![EG\phi]\!]_{U,s})$. Together with the assumption that $s \in [\![\phi]\!]_\varnothing$, we can conclude that $s \in [\![EG\phi]\!]_U$, and hence $\mathcal{M}, s \models_U EG\phi$.

Rule EF_1. We show that $\mathcal{M}, s \models_\varnothing \phi$ implies $\mathcal{M}, s \models_U EF\phi$ when $s \notin U$.

We have the equivalences

$$\begin{aligned}
& s \in [\![EF\phi]\!]_U \\
\overset{(7)}{\Leftrightarrow}\ & s \in \mu Y.([\![\phi]\!]_{\varnothing} \cup pre_{\exists}(Y) \setminus U) \\
\overset{(15)}{\Leftrightarrow}\ & s \in [\![\phi]\!]_{\varnothing} \cup pre_{\exists}(\mu Y.([\![\phi]\!]_{\varnothing} \cup pre_{\exists}(Y) \setminus U \setminus \{s\})) \setminus U \\
\overset{(7)}{\Leftrightarrow}\ & s \in [\![\phi]\!]_{\varnothing} \cup pre_{\exists}([\![EF\phi]\!]_{U,s}) \setminus U
\end{aligned}$$

Now, we can observe that

$$s \in [\![\phi]\!]_{\varnothing} \cup pre_{\exists}([\![EF\phi]\!]_{U,s}) \setminus U$$

holds when $s \in [\![\phi]\!]_{\varnothing}$ and $s \notin U$, and hence $\mathcal{M}, s \models_U EF\phi$.

Rule EF_2. We show that $\mathcal{M}, s' \models_{U,s} EF\phi$ and $s \notin U$ imply $\mathcal{M}, s \models_U EF\phi$. Since we assume $s' \in [\![EF\phi]\!]_{U,s}$ and $s \notin U$, by the same equivalences as in the previous case we can conclude that $s \in [\![EF\phi]\!]_U$, and hence $\mathcal{M}, s \models_U EF\phi$.

Rule AG_1. If $s \in U$, then $s \in [\![AG\phi]\!]_U$ holds by (10).

Rule AG_2. The argument is similar to the proof of EG_2.

Rule AF_1. The argument is similar to the proof of EF_1.

Rule AF_2. The argument is similar to the proof of EF_2.

This concludes the proof of soundness. ■

5 Completeness of the Proof System

A deductive system is termed *complete* if for every semantically valid sequent there exists a derivation of that sequent (that is, all valid sequents can be proved). We show completeness by using the idea of a *canonical proof*. The idea of the proof is that for every valid sequent there is a way to apply rules backwards that is guaranteed to terminate with axiom rules as leaves, and thus, produce a proof of the sequent.

5.1 Reversibility

First, we show that all rules are *reversible*: For each rule, if the conclusion is valid, then there exists a rule that can be applied backward, so that the premises are valid.

Theorem 5.1 (Reversibility)**.** The rules of the proof system from Figure 1 are reversible.

Proof. We consider each rule in turn.

Rule $\wedge$. If $\mathcal{M}, s \models_\varnothing \phi \wedge \psi$ is valid, we can apply Rule $\wedge$ backwards. If $s \in [\![\phi \wedge \psi]\!]_\varnothing$, then necessarily $s \in [\![\phi]\!]_\varnothing$ and $s \in [\![\phi]\!]_\varnothing$, and thus all premises of Rule $\wedge$ are valid, enabling backward application of the rule.

Rules $\vee_1, \vee_2$. If $\mathcal{M}, s \models_\varnothing \phi \vee \psi$ is valid, then $s \in [\![\phi \vee \psi]\!]_\varnothing$, and hence necessarily either $s \in [\![\phi]\!]_\varnothing$ or $s \in [\![\phi]\!]_\varnothing$ has to hold. Thus, either the premises of Rule $\vee_1$ or those of $\vee_2$ are valid and the corresponding rule can be applied backwards.

Rules EX, AX. Assuming $\mathcal{M}, s \models_\varnothing EX\phi$ is valid, then $s \in pre_\exists([\![\phi]\!]_\varnothing)$ by (5). Using the definition of $pre_\exists$ (11), we can conclude that existence of a $s' \in S$ with $s \to s'$ and $s' \in [\![\phi]\!]_\varnothing$ is necessary and thus $\mathcal{M}, s' \models_\varnothing \phi$ is valid. The reasoning for rule AX and $pre_\forall$ is similar.

Rule EG_1. Assuming $\mathcal{M}, s \models_U EG\phi$ and $s \in U$, there is no premise to be proven valid and the rule is always applicable backwards.

Rule EG_2. If we assume $\mathcal{M}, s \models_U EG\phi$, but $s \notin U$, we have to show that $\mathcal{M}, s \models_\varnothing \phi$ and $\mathcal{M}, s' \models_{U,s} EG\phi$. Unfolding the fixed point using Lemma 2.1, we get

$$s \in [\![EG\phi]\!]_U \Leftrightarrow s \in [\![\phi]\!]_\varnothing \cap pre_\exists([\![EG\phi]\!]_{U,s}) \cup U.$$

Since $s \notin U$, then necessarily $s \in [\![\phi]\!]_\varnothing$ and $s \in pre_\exists([\![EG\phi]\!]_{U,s})$. From $s \in pre_\exists([\![EG\phi]\!]_{U,s})$, we can conclude that there exists a s' with $s \to s'$ and $\mathcal{M}, s' \models_{U,s} EG\phi$. $\mathcal{M}, s \models_\varnothing \phi$ follows directly.

Rules EF_1, EF_2. Assuming $\mathcal{M}, s \models_U EF\phi$ and $s \notin U$, we can obtain

$$s \in [\![EF\phi]\!]_U \Leftrightarrow s \in [\![\phi]\!]_\varnothing \cup pre_\exists([\![EF\phi]\!]_{U,s}) \setminus U$$

through unfolding of the fixed point once using Lemma 2.1. Since $s \notin U$, either $s \in [\![\phi]\!]_\varnothing$ or $s \in pre_\exists([\![EF\phi]\!]_{U,s})$ is necessarily valid. From $s \in [\![\phi]\!]_\varnothing$ follows $\mathcal{M}, s \models_\varnothing \phi$. From $s \in pre_\exists([\![EF\phi]\!]_{U,s})$, we can conclude that there exists a s' with $s \to s'$ and $\mathcal{M}, s' \models_{U,s} EF\phi$. Thus, either EF_1 or EF_2 is always applicable backwards when the conclusion is valid.

Rules AG_1, AG_2. The argument is similar to the reversibly of EG_1 and EG_2.

Rules AF_1, AF_2. The argument is similar to the reversibility of EF_1 and EF_2.

This concludes the proof of reversibility. ■

Hence, starting a proof from any semantically valid sequent, there is a way to "grow" a derivation tree upwards, maintaining semantic validity as an invariant property of the nodes of the derivation tree.

5.2 Termination

To obtain a (canonical) proof, however, we need to argue that every branch of the tree is bound to terminate, and furthermore with an axiom.

Lemma 5.1 (Finiteness of Derivation Trees)**.** Every derivation produced with the rules of the proof system from Figure 1 is finite for finite-state Kripke structures.

Proof. Between conclusion and premises, we observe that application of each reversible rule either (i) decreases the length of the sequent formulas or (ii) decreases the number of leftover untagged states $S \setminus U$. Defining a lexicographical ordering through these two criteria on a series of backward applications, it is easy to see that such a series would be monotonically decreasing. ■

5.3 Completeness

Finally, we are ready to show completeness of our proof system.

Theorem 5.2 (Completeness)**.** The proof system from Figure 1 is complete for finite-state Kripke structures.

Proof. For any valid sequent, by Theorem 5.1, there always exists a backwards applicable rule, and, by Lemma 5.1, any series of backward rule applications is terminating. Thus, eventually, every branch must terminate by reverse application of an axiom rule, and hence, there exists a proof of the sequent. ■

Observe that soundness, completeness, and finiteness of derivation trees guarantee *decidability* of sequent validity.

6 Conclusion

In this paper, we have presented a local model-checking proof system for a fragment of CTL, and have proved its soundness and completeness, and termination of proof search for finite-state models. Extending the proof system and the proofs to the full CTL is a routine exercise.

The proof system has been developed for and used in the course *Logic for Computer Scientists*, given at KTH Royal Institute of Technology, Stockholm.

References

[1] Henrik Reif Andersen, Colin Stirling, and Glynn Winskel. A compositional proof system for the modal mu-calculus. In *Proceedings Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 144–153. IEEE, 1994.

[2] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV Version 2: An OpenSource Tool for

Symbolic Model Checking. In *Proc. International Conference on Computer-Aided Verification (CAV 2002)*, volume 2404 of *LNCS*, Copenhagen, Denmark, July 2002. Springer.

[3] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter Kozen, editor, *Logics of Programs, Workshop, Yorktown Heights, New York, USA, May 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[4] Stéphane Demri, Valentin Goranko, and Martin Lange. *Branching-Time Temporal Logics*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

[5] Michael Huth and Mark Dermot Ryan. *Logic in computer science - modelling and reasoning about systems (2. ed.)*. Cambridge University Press, 2004.

[6] Dexter Kozen. Results on the propositional μ-calculus. In Mogens Nielsen and Erik Meineche Schmidt, editors, *Automata, Languages and Programming, 9th Colloquium, Aarhus, Denmark, July 12-16, 1982, Proceedings*, volume 140 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 1982.