Uniform Interpolation Closure for Branching Time Temporal Logics

Renyan Feng[1] , **Yisong Wang**[2,*] , **Mingsen Deng**[1] , **Erman Acar**[3]

[1] Guizhou Provincial Key Laboratory of Computing and Network Convergence, and Guizhou International Science & Technology Cooperation Base of Data Science and Intelligent Decisions (School of Information), Guizhou University of Finance and Economics, China

[2]State Key Laboratory of Public Big Data, Key Laboratory of Advanced Medical Imaging and Intelligent Computing of Guizhou Province, College of Computer Science and Technology, Guizhou University, China

[3]ILLC and IvI, University of Amsterdam, The Netherlands

renyan_feng@163.com, yswang@gzu.edu.cn, msdeng@mail.gufe.edu.cn, e.acar@uva.nl,

Abstract

Computation Tree Logic (CTL) is a fundamental formalism for specifying and reasoning about the behavior of non-terminating systems. Due to the lack of uniform interpolation (UI) property, CTL is usually limited in modular reasoning and system abstraction. This paper studies uniform interpolation in CTL fragments with restricted temporal operators from the point of knowledge forgetting, aiming to identify the minimal logic extensions of these fragments that preserve UI property (referred to as their *uniform interpolation closure*). Our results demonstrate that: (1) CTL(X), the fragment of CTL allowing only the "neXt" operator, enjoys the UI property; (2) the UI closure of the fragment CTL$(\mathrm{F}^{<}, \mathrm{X})$ is the **b**isimulation-invariant fragment of **q**uantified **CTL** in prenex-normal-form, where $\mathrm{F}^{<}$ is the operator "next Future"; (3) CTL$(\mathrm{F}^{<})$ fails to possess Craig interpolation, and every extension of CTL$(\mathrm{F}^{<})$ that possesses the Craig interpolation property is necessarily an extension of CTL(U) which contains only the "Until" operator. These findings provide a precise characterization of the expressive power required to achieve modularity in CTL.

1 Introduction

Temporal logic has established itself as a cornerstone of formal reasoning, providing the essential machinery for system specification and verification [Wolper, 1985; Bradfield and Walukiewicz, 2018]. Among various formalisms, Computation Tree Logic (CTL) stands out as a prominent branching-time logic, playing a pivotal role in model checking and the verification of concurrent systems [Clarke *et al.*, 1986].

In model checking and specification engineering, one often needs to eliminate irrelevant propositions to obtain an abstraction or projection over a smaller vocabulary. A logic satisfies the property of *uniform interpolation* (UI) if it is functionally closed under such propositional eliminations. The significance of UI spans across foundational logical results, such as Beth's definability [Beth, 1953], and core Knowledge Representation and Reasoning (KR) tasks, including knowledge extraction [Toluhi, 2023] and critical verification methods [Gianola, 2023].

A dual and equally vital notion to UI is *forgetting* [Konev *et al.*, 2008; Toluhi, 2023]. Forgetting eliminates redundant knowledge or attributes from a knowledge base while strictly preserving its essential logical properties. As a fundamental method for modularity and knowledge extraction in KR [Eiter and Kern-Isberner, 2019; Gonçalves *et al.*, 2023], it has been persistently and extensively studied across diverse logic systems. This includes foundational research in classical logic [Lin and Reiter, 1994; Doherty *et al.*, 2001; Marquis, 2003; Zhou and Zhang, 2011; Doherty and Szalas, 2024; Liberatore, 2024; Sauerwald *et al.*, 2024], description logic [Wang *et al.*, 2010; Lutz and Wolter, 2011; Zhao *et al.*, 2020], temporal logic CTL and μ-calculus [Feng *et al.*, 2020; Feng *et al.*, 2022; Feng *et al.*, 2023], and (multi-)modal logic [Zhang and Foo, 2006; Su *et al.*, 2009; Fang *et al.*, 2019]. Furthermore, forgetting has been applied to dynamic frameworks such as situation calculus [Lin and Reiter, 1997; Liu and Claßen, 2025] and answer set programming [Eiter and Wang, 2008; Wang *et al.*, 2014].

While recent research has systematically investigated the existence of interpolants across major non-propositional logics—including first-order theories [Calvanese *et al.*, 2021; Calvanese *et al.*, 2022], modal logic [Herzig and Mengin, 2008; Férée *et al.*, 2024; Ghilardi and Zawadowski, 2013], first-order modal logic [Kurucz *et al.*, 2023], linear logic [Gheerbrant and ten Cate, 2009; Fussner and Santschi, 2024], and (modal) μ-calculus [D'Agostino and Hollenberg, 1996; D'Agostino and Hollenberg, 2000; D'Agostino and Lenzi, 2006; Afshari *et al.*, 2021]—a seminal observation in temporal logic research is that Craig interpolation often fails for standard systems such as linear temporal logic (LTL), CTL, and CTL* [Maksimova, 1991; D'Agostino, 2008]. Demonstrating CTL's expressive gap due to its lack of the Beth property, we adapt an example from [Maksimova, 1991]:

Example 1. *Let p and x be two distinct propositions, and let $\phi(p, x)$[1] be the conjunction of the following CTL formulas:*

$$\mathrm{AG}(\neg x \wedge \neg \mathrm{AG}p \rightarrow \neg \mathrm{AX}\neg x), \qquad \mathrm{AG}(\neg \mathrm{AX}\neg x \rightarrow \mathrm{AX}x),$$
$$\mathrm{AG}(\mathrm{AX}x \rightarrow \neg x \wedge \neg \mathrm{AG}p), \quad \mathrm{AG}(x \rightarrow \neg \mathrm{AG}p), \quad \mathrm{AG}(\mathrm{AFAG}p).$$

[1] $\phi(p, x)$ means the formula ϕ with $\mathrm{var}(\phi) = \{p, x\}$.

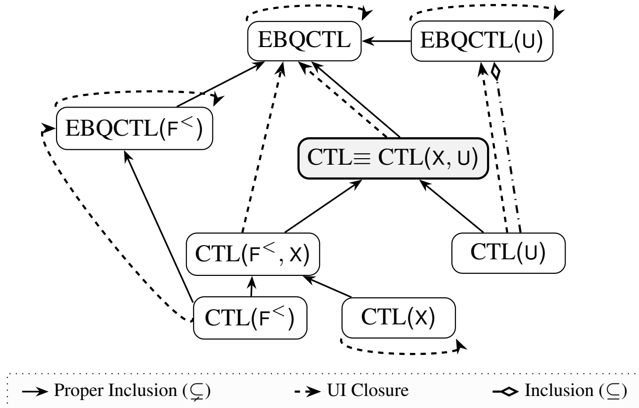

Figure 1: Uniform Interpolation Closures for CTL Fragments. The inclusion and proper inclusion relations depicted herein refer to the relative expressive power between the respective logical languages.

It can be shown that $\phi(p,x)\wedge\phi(p,y) \models x \leftrightarrow y$, which semantically defines x in terms of p. Here, $\phi(p,y)$ is obtained from $\phi(p,x)$ by replacing all occurrences of x with y. However, there is no CTL formula φ over the signature $\{p\}$ such that $\phi(p,x) \models x \leftrightarrow \varphi$. This is because the implicit definability of x relies on long-range dependencies and counting properties that exceed the expressive power of CTL, thus precluding its explicit definition.

This negative result raises two fundamental questions: (1) Which logical language effectively describes such uniform interpolants? and (2) What constitutes the UI closure—defined as the minimal extension of the logic that preserves the UI property—for CTL fragments. Answering these questions delineates exactly when CTL level automation can be preserved and when one must escalate to a stronger formalism.

In this paper, we investigate the UI closures of key CTL fragments through the lens of knowledge forgetting, employing inductive proofs and constructive methods. We focus on fragments defined over subsets of temporal operators $S \subseteq \{\mathrm{X}, \mathrm{F}^{<}, \mathrm{U}\}$. In particular, for a given temporal logic language $\mathcal{L}$ the formulas in $\mathcal{L}(S)$ only contain the temporal operators in S. Our main contributions are summarized in the landscape presented in Figure 1. Specifically, we identify EBQCTL(the **b**isimulation-invariant fragment of **q**uantified **CTL** (QCTL) in prenex normal form [Laroussinie and Markey, 2014]) as a robust candidate for the UI closure of several key fragments. This mapping not only clarifies the relative expressive power of these logics but also pinpoints where propositional quantification becomes necessary to achieve closure under forgetting.

2 Preliminaries

In this section, we present the pivotal notions and notations central to our work.

2.1 Temporal Languages

A *frame* $\mathcal{T} = (S, R)$ comprises a nonempty set of worlds (states) S and a binary relation $R \subseteq S \times S$. A *path* π is an infinite sequence $s_0 s_1 s_2 \ldots$ satisfying $(s_j, s_{j+1}) \in R$ for all $j \geq 0$. We denote $s \in \pi$ if state s occurs in π, and π_s as a path starting at s. A state s' is *reachable* from state s if s' appears in π_s. A state s is *initial* iff every $s' \in S$ is reachable from s.

A *propositional signature* σ is a finite nonempty set $\{p_1, \ldots, p_n\}$ of atomic propositions. A *point σ-structure* is a triple $\mathcal{M} = (\mathcal{T}, L, s)$ where: $\mathcal{T} = (S, R)$ is a frame and R is serial ($\forall s \in S, \exists s' \in S : (s, s') \in R$); $s \in S$ is a state; and $L : S \to \mathscr{P}(\sigma)$ (or 2^σ) is a labeling function.

An *initial σ-structure* is a point σ-structure $\mathcal{M} = (\mathcal{T}, L, s_0)$ where s_0 is an initial state in $\mathcal{T}$. An *initial (resp. point) structure* is an initial (resp. point) σ-structure for some signature σ. Given a σ-structure $\mathcal{M} = (\mathcal{T}, L, s)$, the terms "path", "reachable", and "initial" in the context of $\mathcal{M}$ retain the same meaning as they do within the frame $\mathcal{T}$.

Given a signature V, two initial structures $\mathcal{M}_1$ and $\mathcal{M}_2$ are *V-bisimilar* (denoted $\mathcal{M}_1 \leftrightarrow_V \mathcal{M}_2$) iff there exists a V-bisimulation relation such that: (1) their initial states are related, and (2) all subsequent states can mutually simulate each other's behavior when disregarding the atoms in V (see [Feng *et al.*, 2022] for details).

The collection of all initial σ-structures is denoted by $Str[\sigma]$. For $\sigma \subseteq \tau$, the σ-reduct $\mathcal{M} \restriction \sigma$ of an initial τ-structure $\mathcal{M} = (\mathcal{T}, L, s)$ (with $\mathcal{T} = (S, R)$) is defined as $(\mathcal{T}, L \restriction \sigma, s)$, where $L \restriction \sigma(s') = L(s') \cap \sigma$ for each state $s' \in S$. We say $\mathcal{M}$ is a τ-expansion of $\mathcal{M} \restriction \sigma$. For any $K \subseteq Str[\tau]$, we define $K \restriction \sigma = \{\mathcal{M} \restriction \sigma \mid \mathcal{M} \in K\}$ and $\overline{K} = Str[\tau] \setminus K$. It is obvious that $\mathcal{M} \leftrightarrow_{\tau\setminus\sigma} \mathcal{M} \restriction \sigma$.

Given an initial σ-structure $(\mathcal{T}, L, s)$, a state set $A \subseteq S$, and a fresh propositional letter p, we define $L[A/p]$ as

$$L[A/p](s') = \begin{cases} L(s') \cup \{p\}, & \text{if } s' \in A, \\ L(s'), & \text{otherwise,} \end{cases}$$

and the corresponding $\sigma \cup \{p\}$-expansion as the structure $(\mathcal{T}, L[A/p], w)$. Furthermore, for any formula φ and proposition p, let $\varphi[p/\psi]$ denote the formula obtained by substituting all occurrences of p in φ with ψ.

Definition 1 (Temporal language [Gheerbrant and ten Cate, 2009]). *A temporal language is a triple $\mathcal{L} = (\mathcal{F}, \mathcal{K}, \models_{\mathcal{L}})$, where $\mathcal{F}$ is a map from propositional signatures to sets of formulas, $\mathcal{K}$ is a set of initial structures, and $\models_{\mathcal{L}}$ is the satisfiability relation between initial structures and formulas, satisfying the following conditions, for all propositional signatures σ, τ:*

(i) Expansion property. If $\sigma \subseteq \tau$ then $\mathcal{F}(\sigma) \subseteq \mathcal{F}(\tau)$. Furthermore, for all $\varphi \in \mathcal{F}(\sigma)$ and $\mathcal{M} \in Str[\tau]$, $\mathcal{M} \models_{\mathcal{L}} \varphi$ iff $\mathcal{M} \restriction \sigma \models_{\mathcal{L}} \varphi$. If $\mathcal{M} \in Str[\sigma]$ and $\mathcal{M} \models_{\mathcal{L}} \varphi$, then $\varphi \in \mathcal{F}(\sigma)$.

(ii) Closure under uniform substitution. For all $\psi \in \mathcal{F}(\sigma)$, $p \notin \sigma$, and $\varphi \in \mathcal{F}(\sigma \cup \{p\})$, there is a formula of $\mathcal{F}(\sigma)$, which we will denote by $\varphi[p/\psi]$, such that for every $(\mathcal{T}, L, w) \in Str[\sigma]$ the following holds:

$$(\mathcal{T}, L, w) \models_{\mathcal{L}} \varphi[p/\psi] \text{ iff } (\mathcal{T}, L', w) \models_{\mathcal{L}} \varphi$$

where $L' = L[\{w' \mid (\mathcal{T}, L, w') \models_{\mathcal{L}} \psi\}/p]$.

(iii) Negation property. For each $\varphi \in \mathcal{F}(\sigma)$ there is a formula of $\mathcal{L}[\sigma]$, which we will denote $\neg\varphi$, s.t. for all $\mathcal{M} \in Str[\sigma]$, $\mathcal{M} \models_{\mathcal{L}} \neg\varphi$ iff $\mathcal{M} \not\models_{\mathcal{L}} \varphi$.

Given a propositional signature σ, the temporal language over σ is denoted by $\mathcal{L}[\sigma]$, i.e., $\mathcal{F}$ is a map with σ as its domain and $\mathcal{K} = Str[\sigma]$. Given a formula φ, by $\varphi \in \mathcal{L}[\sigma]$, we mean that $\varphi \in \mathcal{F}(\sigma)$. $\varphi \in \mathcal{L}$ indicates that φ in $\mathcal{L}[\sigma]$ for some signature σ. The temporal language $\mathcal{L}$ is *bisimulation-invariant* if the following condition holds: for all signatures σ, for each formula $\varphi \in \mathcal{L}[\sigma]$, and for all structures $\mathcal{M}, \mathcal{M}' \in Str[\sigma]$, if $\mathcal{M} \leftrightarrow_\emptyset \mathcal{M}'$, then $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models \varphi$.

For any $\varphi \in \mathcal{L}[\sigma]$, we define $\mathsf{Mod}^\sigma(\varphi) = \{\mathcal{M} \in Str[\sigma] \mid \mathcal{M} \models_\mathcal{L} \varphi\}$ (subscript/superscript omitted when clear from context). $\mathcal{M}$ is called a *model* of φ when $\mathcal{M} \models \varphi$. If $\mathsf{Mod}(\varphi) \neq \emptyset$, then φ is *satisfiable*. φ is *valid* if for every initial structure $\mathcal{M}$, there is $\mathcal{M} \models \varphi$. A model is *finite* whenever it contains a finite set of worlds; otherwise it is *infinite*. A class $K \subseteq Str[\sigma]$ is *definable* in the $\mathcal{L}[\sigma]$ if there exists a $\mathcal{L}[\sigma]$-formula φ such that $\mathsf{Mod}(\varphi) = K$. Moreover, given two formulas $\varphi, \psi \in \mathcal{L}[\sigma]$, φ entails ψ, denoted as $\varphi \models \psi$, if $\mathsf{Mod}(\varphi) \subseteq \mathsf{Mod}(\psi)$. If $\mathsf{Mod}(\varphi) = \mathsf{Mod}(\psi)$, then φ is equivalent to ψ, denoted $\varphi \equiv \psi$.

For a given formula φ, $\mathrm{var}(\varphi)$ denotes the set of atoms occurring in φ. And for any signature V, φ is V-irrelevant, denoted as $IR(\varphi, V)$, whenever there exists a formula ψ s.t. $\mathrm{var}(\psi) \cap V = \emptyset$ and $\mathsf{Mod}(\psi) = \mathsf{Mod}(\varphi)$. Moreover, $|V|$ denotes the number of atoms in V and $|\varphi|$ represents the size of φ (measured by the number of symbols).

Definition 2 (Extension of a temporal language)**.** *For temporal languages $\mathcal{L}_1 = (\mathcal{F}_1, \mathcal{K}_1, \models_{\mathcal{L}_1})$ and $\mathcal{L}_2 = (\mathcal{F}_2, \mathcal{K}_2, \models_{\mathcal{L}_2})$, we say that $\mathcal{L}_2$ extends $\mathcal{L}_1$ (denoted $\mathcal{L}_1 \subseteq \mathcal{L}_2$) if:*

$$\forall\sigma\forall\varphi \in \mathcal{L}_1[\sigma]\ \exists\varphi' \in \mathcal{L}_2[\sigma]\ s.t.\ \mathsf{Mod}(\varphi) = \mathsf{Mod}(\varphi').$$

In this case, $\mathcal{L}_1$ is called a fragment of $\mathcal{L}_2$.

Two temporal languages $\mathcal{L}_1$ and $\mathcal{L}_2$ have *the same expressive power*, denoted as $\mathcal{L}_1 =_e \mathcal{L}_2$, whenever $\mathcal{L}_1 \subseteq \mathcal{L}_2$ and $\mathcal{L}_2 \subseteq \mathcal{L}_1$.

We now recall the notions of projective class and uniform interpolation.

Definition 3 (Projective class [Gheerbrant and ten Cate, 2009])**.** *Let σ be a propositional signature, $\mathcal{L}$ be a temporal language, and $K \subseteq Str[\sigma]$. We say that K is a projective class of $\mathcal{L}$ if there exists a signature $\tau \supseteq \sigma$ and a formula $\varphi \in \mathcal{L}[\tau]$ such that $K = \mathsf{Mod}(\varphi) \upharpoonright \sigma$.*

Intuitively, a projective class comprises the set of σ-reducts of the models of a formula ϕ over $\mathcal{L}[\tau]$, where $\sigma \subseteq \tau$ (refer to Example 3 in Section 3 for an illustration). The following lemma establishes that any projective class defined over a temporal language remains a projective class within its extension.

Lemma 1. *If $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then every projective class of $\mathcal{L}_1$ is also a projective class of $\mathcal{L}_2$.*

The uniform interpolant was originally defined by D'Agostino and Hollenberg (1996) as follows:

Definition 4 (Uniform interpolant)**.** *Let $\mathcal{L}$ be a temporal language, σ and τ be signatures with $\sigma \subseteq \tau$, and $\varphi \in \mathcal{L}[\tau]$. ψ is a uniform interpolant of φ with respect to σ if the following conditions are satisfied:*

- *$var(\psi) \subseteq \sigma$;*
- *$\varphi \models \psi$; and*
- *for every formula $\alpha \in \mathcal{L}[\tau']$ with $\tau \cap \tau' \subseteq \sigma$, there is $\varphi \models \alpha$ iff $\psi \models \alpha$.*

$\mathcal{L}$ has the *uniform interpolation* (UI) property if, for all signatures $\sigma \subseteq \tau$ and for each formula $\varphi \in \mathcal{L}[\tau]$ there is a uniform interpolant of φ with respect to σ.

Definition 5 (Interpolant)**.** *Let $\mathcal{L}$ be a temporal language, τ and τ' be signatures, $\varphi \in \mathcal{L}[\tau]$, $\psi \in \mathcal{L}[\tau']$, and $\varphi \models \psi$. A formula $\theta \in \mathcal{L}[\tau \cap \tau']$ is called an interpolant of φ and ψ iff $\varphi \models \theta$ and $\theta \models \psi$.*

A temporal language $\mathcal{L}$ has Craig's interpolation property whenever there is an interpolant for every formula φ and ψ with $\varphi \models \psi$. Craig's interpolation is a weaker variant of the UI property.

2.2 CTL and QCTL

Let σ be a propositional signature. The *formula* ϕ of temporal language QCTL over σ is defined as

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \exists p.\phi \mid \mathrm{EX}\phi \mid \mathrm{EF}\phi \mid \mathrm{EF}^{<}\phi \mid \mathrm{E}(\phi\ \mathrm{U}\ \phi) \tag{1}$$

where $p \in \sigma$; E (spoken as "there exists a path") is a path quantifier; $\exists$ (spoken as "there exists a proposition") is a proposition quantifier; X, F, $\mathrm{F}^{<}$ and U are temporal operators meaning 'neXt state', 'some Future state', 'next Future' and 'Until', respectively. The symbols $\top$ and $\bot$ are defined as abbreviations for the formulas $p \vee \neg p$ and $p \wedge \neg p$, respectively. The formulas $\phi \wedge \psi$, $\phi \to \psi$, and $\phi \leftrightarrow \psi$ stand for $\neg(\neg\phi \vee \neg\psi)$, $\neg\phi \vee \psi$, and $(\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)$. Let $P = \{p_1, \ldots, p_n\}$ and φ be a QCTL formula. $\exists P.\varphi$ stands for the QCTL formula $\exists p_1.(\exists p_2.(\ldots(\exists p_n.\varphi)\ldots))$. The class of QCTL formulas without propositional quantifiers coincides with CTL formulas.

Given a QCTL formula φ and a point structure $\mathcal{M} = (\mathcal{T}, L, s)$ with $\mathcal{T} = (S, R)$, the relation $(\mathcal{T}, L, s) \models \varphi$, called $(\mathcal{T}, L, s)$ satisfies φ, is recursively defined as follows:

- $(\mathcal{T}, L, s) \models p$ if and only if $p \in L(s)$;
- $(\mathcal{T}, L, s) \models \neg\varphi$ if and only if $(\mathcal{T}, L, s) \not\models \varphi$;
- $(\mathcal{T}, L, s) \models \varphi_1 \vee \varphi_2$ if and only if $(\mathcal{T}, L, s) \models \varphi_1$ or $(\mathcal{T}, L, s) \models \varphi_2$;
- $(\mathcal{T}, L, s) \models \exists p.\varphi$ if and only if there is $(\mathcal{T}, L', s)$ with L' different from L only on p such that $(\mathcal{T}, L', s) \models \varphi$;
- $(\mathcal{T}, L, s) \models \mathrm{EX}\varphi$ if and only if there exists $(s, s') \in R$ s.t. $(\mathcal{T}, L, s') \models \varphi$;
- $(\mathcal{T}, L, s) \models \mathrm{EF}\varphi$ if and only if there is a path $\pi = ss_1s_2\ldots$ of $\mathcal{T}$ s.t. $(\mathcal{T}, L, s') \models \varphi$ for some $s' \in \pi$;
- $(\mathcal{T}, L, s) \models \mathrm{EF}^{<}\varphi$ if and only if there exists a path $\pi = s_0s_1s_2\ldots$ of $\mathcal{T}$, with $s = s_0$, s.t. at least one state s_i ($i > 0$) along π satisfies $(\mathcal{T}, L, s_i) \models \varphi$;
- $(\mathcal{T}, L, s) \models \mathrm{E}(\varphi_1 \mathrm{U} \varphi_2)$ if and only if there exists $k \geq 1$ and a path $s_1s_2\ldots$ of $\mathcal{T}$, with $s_1 = s$, s.t. $(\mathcal{T}, L, s_k) \models \varphi_2$, and $(\mathcal{T}, L, s_j) \models \varphi_1$ for every j $(1 \leq j < k)$.

One can check that $\mathrm{EF}^{<}\varphi \equiv \mathrm{EXEF}\varphi$. Let A be the path quantifier that is dual to E; $\forall$ be the proposition quantifier dual to $\exists$; G and $\mathrm{G}^{<}$ are shorthand for $\neg\mathrm{F}\neg$ and $\neg\mathrm{F}^{<}\neg$, respectively.

The other form formulas of QCTL$[\sigma]$ can be defined using formulas in the forms of (1).

Let $A \subseteq \{\mathrm{X}, \mathrm{F}, \mathrm{F}^{<}, \mathrm{U}\}$. The notation $\mathcal{L}(A)$ denotes the fragment of temporal language $\mathcal{L}$ using only the temporal operators in A. Indeed, QCTL $=_e$ QCTL(A) when $A = \{\mathrm{X}, \mathrm{F}, \mathrm{F}^{<}, \mathrm{U}\}$ or $A = \{\mathrm{X}, \mathrm{F}^{<}, \mathrm{U}\}$.

A significant fragment of QCTL is BQCTL, characterized as the bisimulation-invariant subset of QCTL [Rohde, 2002; Colcombet *et al.*, 2024]. Since QCTL matches the expressive power of Monadic Second-Order Logic (MSO) (cf. Proposition 3.4 of [Laroussinie and Markey, 2014]), BQCTL corresponds to the bisimulation-invariant fragment of MSO, which is equivalent to the μ-calculus (cf. Theorem 14.18 of [Rohde, 2002]). Consequently, for any $\varphi \in$ BQCTL and $\mathcal{M}, \mathcal{M}' \in Str[\mathrm{var}(\varphi)]$, $\mathcal{M} \leftrightarrow_{\emptyset} \mathcal{M}'$ entails $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models \varphi$. Furthermore, $\mathcal{M} \models \exists p.\varphi$ holds if there is $\mathcal{M}'$ such that $\mathcal{M} \leftrightarrow_{\{p\}} \mathcal{M}'$ and $\mathcal{M}' \models \varphi$. We also introduce EBQCTL, defined as the bisimulation-invariant subclass of prenex QCTL formulas [2]. We demonstrate that EBQCTL and BQCTL are equally expressive.

Lemma 2. *EBQCTL and BQCTL are equally expressive.*

Recall that μ-calculus possesses the finite model property [Bradfield and Stirling, 2007] and BQCTL $=_e$ μ-calculus. The following result holds.

Proposition 1. *Given two satisfiable BQCTL formulas φ_1 and φ_2, if φ_1 and φ_2 have the same finite models, then they have the same infinite models as well.*

Proof. (Sketch.) If φ_1 and φ_2 have the same finite models, then for any finite model $\mathcal{N}$, $\mathcal{N}$ satisfies or does not satisfy both φ_1 and φ_2 simultaneously.

Suppose there is some infinite initial structure $\mathcal{M}$ such that $\mathcal{M}$ is a model of φ_1, but not the model of φ_2. Therefore, $\mathcal{M}$ is a model of $\varphi_1 \wedge \neg\varphi_2$, i.e., $\varphi_1 \wedge \neg\varphi_2$ is satisfiable. Then $\varphi_1 \wedge \neg\varphi_2$ has a finite model $\mathcal{M}'$ according to finite model property. One could note that $\mathcal{M}' \models \varphi_1$, and hence $\mathcal{M}' \models \varphi_2$, which contradicts with $\mathcal{M}' \models \neg\varphi_2$. Therefore, φ_1 and φ_2 have the same infinite models. □

By Proposition 1, two BQCTL formulas are logically equivalent if and only if they share the same set of finite models. Consequently, without loss of generality, we restrict our focus to finite initial structures throughout this paper.

3 Forgetting in BQCTL

This section extends the notion of forgetting from CTL [Feng *et al.*, 2023] to BQCTL and investigates the intrinsic connections among forgetting, projective classes, and UI within the BQCTL framework.

Definition 6 (Forgetting)**.** *Let φ be a BQCTL formula, $P \subseteq var(\varphi)$, and $\sigma = var(\varphi) \setminus P$. A formula ψ is a* result of forgetting P from φ*, if*

$$\mathsf{Mod}(\psi) = \{\mathcal{M} \in Str[\sigma] \mid \exists \mathcal{M}' \in \mathsf{Mod}(\varphi) \textit{ and } \mathcal{M} \leftrightarrow_P \mathcal{M}'\}. \quad (2)$$

[2]Syntactically, this fragment comprises BQCTL formulas where all propositional quantifiers are scoped outside the CTL components.

Note that the result of forgetting is unique up to logical equivalence, and is denoted by *forget*(φ, P). For any BQCTL formula φ, the expression $\exists p.\varphi$ represents the result of forgetting the atom p from φ. Leveraging the property that *forget*$(\varphi, P \cup \{p\}) =$ *forget*(*forget*$(\varphi, \{p\}), P)$ [Feng *et al.*, 2023] and the expressive equivalence BQCTL $=_e$ μ-calculus, we extend this notation and use $\exists P.\varphi$ to denote the result of forgetting a set of atoms P from φ.

Example 2. *Let $\varphi = \mathrm{EX}(p \wedge q)$ be a BQCTL formula, $V = \{p\}$, and $\sigma = var(\varphi) \setminus V$.*

According to the semantics of $\exists p.\mathrm{EX}(p \wedge q)$, we have:

$$\mathsf{Mod}(\exists p.\mathrm{EX}(p \wedge q)) =$$
$$\{\mathcal{M} \in Str[\sigma] \mid \exists \mathcal{M}' \in \mathsf{Mod}(\varphi) \textit{ and } \mathcal{M} \leftrightarrow_V \mathcal{M}'\}.$$

Therefore, $\exists p.\mathrm{EX}(p \wedge q)$ is the result of forgetting V from φ according to Definition 6.

By Definition 3, a class $K \subseteq Str[\sigma]$ being a projective class of BQCTL implies that it is the set of models of the result of forgetting $\tau \setminus \sigma$ ($\sigma \subseteq \tau$) from some formula $\varphi \in$ BQCTL$[\tau]$. Formally:

Proposition 2. *Let $K \subseteq Str[\sigma]$ and $\varphi \in BQCTL[\tau]$ ($\sigma \subseteq \tau$) be a formula that makes K a projective class of BQCTL, i.e., $K = \mathsf{Mod}(\varphi) \restriction \sigma$. We have $\mathsf{Mod}(\varphi) \restriction \sigma = \mathsf{Mod}(\exists P.\varphi)$ with $P = \tau \setminus \sigma$.*

Example 3. *Let $\sigma = \{q\}$, K be the set of initial σ-structures where q holds in all immediate successors of the initial state, and $\varphi = \mathrm{AX}(p \wedge q)$ be a CTL formula over $\tau = \{p, q\}$.*

It is apparent that:

$$\mathsf{Mod}(\varphi) = \{\mathcal{M} = ((S, R), L, s_0) \in Str[\tau] \mid \forall (s_0, s_1) \in R:\ ((S, R), L, s_1) \models p \wedge q\};$$
$$\mathsf{Mod}(\varphi) \restriction \sigma = \{\mathcal{M} \restriction \sigma \mid \mathcal{M} \in \mathsf{Mod}(\varphi)\} = K;$$
$$\mathsf{Mod}(\exists P.\varphi) = \{\mathcal{M} \in Str[\sigma] \mid \exists \mathcal{M}' \in \mathsf{Mod}(\varphi) \textit{ and } \mathcal{M} \leftrightarrow_P \mathcal{M}'\}, \textit{ where } P = \tau \setminus \sigma.$$

Thus, K is a projective class of CTL and $\mathsf{Mod}(\varphi) \restriction \sigma = \mathsf{Mod}(\exists P.\varphi)$.

On the one hand, according to Theorem 5 of [Feng *et al.*, 2023], given a signature τ, we have $\varphi \models$ *forget*(φ, P) and *IR*(*forget*$(\varphi, P), P)$ for any formula $\varphi \in$ BQCTL$[\tau]$ and for any $P \subseteq \mathrm{var}(\varphi)$. On the other hand, for every formula $\psi \in$ BQCTL$[\tau']$ with $\tau \cap \tau' \subseteq \mathrm{var}($*forget*$(\varphi, P))$, if $\varphi \models \psi$ then *forget*$(\varphi, P) \models \psi$. This establishes forgetting as the semantic counterpart of UI. Formally:

Lemma 3. *Let σ, τ be signatures with $\sigma \subseteq \tau$, φ in BQCTL$[\tau]$, and $P = \tau \setminus \sigma$. Then $\exists P.\varphi \in BQCTL[\sigma]$ is the result of forgetting P from φ iff it is the uniform interpolant of φ with respect to σ.*

Lemma 3 demonstrates that a logic $\mathcal{L}$ is closed under forgetting—that is, the result of forgetting any set of propositions in any formula of $\mathcal{L}$ is always expressible within $\mathcal{L}$—if and only if $\mathcal{L}$ possesses the UI property.

4 Temporal Languages with Uniform Interpolation

In this section, we prove that several fragments of BQCTL enjoy the UI property. Moreover, we provide an algorithm for computing the uniform interpolants of formulas in the CTL(X) fragment.

4.1 UI for Fragments of BQCTL

As BQCTL extends CTL with the second-order quantifiers, the following theorem can be directly derived.

Theorem 1. *Let $A \subseteq \{\mathrm{X}, \mathrm{F}, \mathrm{F}^{<}, \mathrm{U}\}$. Then BQCTL($A$) and EBQCTL($A$) have the UI property.*

Proof. Let σ, τ with $\sigma \subseteq \tau$ be signatures, and $P = \tau \setminus \sigma$.

On the one hand, it is obvious that $\exists P.\phi$ in BQCTL$[\sigma]$ for any formula $\phi \in$ BQCTL, and it is the result of forgetting P from φ by Definition 6. On the other hand, according to Lemma 3, $\exists P.\phi$ is the uniform interpolant of ϕ with respect to σ.

More generally, $\exists P.\varphi$ is the result of forgetting P from φ for any formula φ in BQCTL$(A)[\tau]$ (resp., EBQCTL$(A)[\tau]$) , and $\exists P.\varphi$ is in BQCTL$(A)[\sigma]$ (resp., EBQCTL$(A)[\sigma]$).

Therefore, BQCTL(A) (resp., EBQCTL(A)) has the UI property. $\square$

Theorem 1 shows that, for any $A \subseteq \{\mathrm{X}, \mathrm{F}, \mathrm{F}^{<}, \mathrm{U}\}$, the forgetting operator remains closed within the fragments BQCTL(A) and EBQCTL(A) of BQCTL and EBQCTL, respectively.

4.2 UI for CTL(X)

Férée et al. (2024) established the UI property for modal logic K using sequent calculi. Given that CTL(X) is a variant of K sharing the same expressive power, it inherently inherits this property. Departing from these K-based approaches, we provide a CTL-native constructive treatment stated directly within the branching-time setting. Specifically, we conduct a structural analysis of CTL(X) formulas to facilitate a computable algorithm. This result serves not merely as an existential proof, but as a logic-specific component of our broader closure classification across CTL fragments. We hereby define the nesting depth of a proposition p in a CTL(X) formula φ with respect to the temporal operator X.

Definition 7. *Let $\varphi \in$ CTL(X) and let p be a proposition occurring in φ. The nesting depth of p in φ with respect to X (referred to as "the X-depth of p in φ"), denoted by $dep_{\mathrm{X}}(\varphi, p)$, is defined inductively as follows:*

$$
\begin{aligned}
dep_{\mathrm{X}}(p,p) &= 0;\\
dep_{\mathrm{X}}(\varphi_1 \vee \varphi_2, p) &= dep_{\mathrm{X}}(\varphi_1 \wedge \varphi_2, p)\\
&= \max\{dep_{\mathrm{X}}(\varphi_1, p), dep_{\mathrm{X}}(\varphi_2, p)\};\\
dep_{\mathrm{X}}(\neg\varphi_1, p) &= dep_{\mathrm{X}}(\varphi_1, p);\\
dep_{\mathrm{X}}(\mathrm{EX}\varphi_1, p) &= dep_{\mathrm{X}}(\mathrm{AX}\varphi_1, p) = 1 + dep_{\mathrm{X}}(\varphi_1, p).
\end{aligned}
$$

In the following, supposing all formulas in CTL(X) are in negation normal form (NNF) [3].

Example 4. *Given $\varphi_1 = \mathrm{EX}p$, $\varphi_2 = \mathrm{EX}(q \wedge \mathrm{EX}(p \vee q))$, $\varphi_3 = \mathrm{EX}(p \wedge f) \wedge \mathrm{EX}(\neg p \wedge r)$, and $\varphi' = \varphi_1 \wedge \varphi_2$, we have $dep_{\mathrm{X}}(\varphi_1, p) = 1$, $dep_{\mathrm{X}}(\varphi_2, p) = 2$, $dep_{\mathrm{X}}(\varphi_3, p) = 1$, and $dep_{\mathrm{X}}(\varphi', p) = 2$.*

[3] A formula φ is said to be in negation normal form (NNF) whenever the negations are only applied to propositions in φ.

Algorithm 1 AL-Forget(φ, V)

Require: A CTL(X) formula φ and a signature V
Ensure: The result of forgetting V from φ

if φ is unsatisfiable **then**
  return $\bot$
end if
for all $p \in V$ **do**
  $E \leftarrow \{\langle 0 \sim 0\rangle\} \cup \{\langle n \sim i\rangle \mid 1 \leq n \leq dep_{\mathrm{X}}(\varphi, p)$ and $1 \leq i \leq \text{Num-EX}_n(\varphi, p)\}$
  $\psi \leftarrow \bigvee_{S \subseteq E} \varphi_{\mathcal{H},p}[p^i/\top]_{i \in S}[p^i/\bot]_{i \in E \setminus S}$
  $\varphi \leftarrow \mathbf{Rem}(\psi)$
end for
return φ

Consider the formula $\varphi_3 = \mathrm{EX}(p \wedge f) \wedge \mathrm{EX}(\neg p \wedge r)$. Observe that there are two distinct occurrences of the operator EX at the same X-depth relative to p. To facilitate the construction of uniform interpolants for CTL(X), we introduce the following notations to distinguish these occurrences:

- Num-$\mathrm{EX}_n(\varphi, p)$: Denotes the total number of EX operator instances occurring at the X-depth n of p within φ. For example, in the formula φ_3 above, Num-$\mathrm{EX}_1(\varphi_3, p) = 2$.
- **Dis-EX**(φ, p): A labeled version of φ where distinct occurrences of EX at the same depth are uniquely indexed. Specifically, if there are m instances of EX at the X-depth n associated with p, we index them sequentially. The i-th occurrence (where $1 \leq i \leq m$) is denoted by $\mathrm{E}_{\langle n \sim i\rangle}\mathrm{X}$.
- $\varphi_{\mathcal{H},p}$: Represents the fully annotated formula obtained by indexing occurrences of the atom p within **Dis-EX**(φ, p). Let p be an atom occurring at X-depth n (where $0 \leq n \leq dep_{\mathrm{X}}(\varphi, p)$). We denote this occurrence as: $p^{\langle 0 \sim 0\rangle}$, if it does not fall within the scope of any indexed operator $\mathrm{E}_{\langle n \sim i\rangle}\mathrm{X}$; $p^{\langle n \sim i\rangle}$, if it appears within the scope of $\mathrm{E}_{\langle n \sim i\rangle}\mathrm{X}$, where $1 \leq i \leq$ Num-$\mathrm{EX}_n(\varphi, p)$.

Example 5 (cont'd of Example 4)**.** *Let $\varphi = \varphi_2 \wedge \varphi_3$. Then, the following holds:*

$$
\begin{aligned}
&\textit{Num-}\mathrm{EX}_1(\varphi, p) = 3; \textit{Num-}\mathrm{EX}_2(\varphi, p) = 1\\
&\varphi_{\mathcal{H},p} = (\textbf{\textit{Dis-}}\mathbf{EX}(\varphi_2 \wedge \varphi_3))_{\mathcal{H},p}\\
\equiv&(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(q \wedge \mathrm{E}_{\langle 2\sim 1\rangle}\mathrm{X}(p \vee q)) \wedge\\
&\mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(p \wedge f) \wedge \mathrm{E}_{\langle 1\sim 3\rangle}\mathrm{X}(\neg p \wedge r))_{\mathcal{H},p}\\
\equiv&(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(q \wedge \mathrm{E}_{\langle 2\sim 1\rangle}\mathrm{X}(p^{\langle 2\sim 1\rangle} \vee q)) \wedge\\
&\mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(p^{\langle 1\sim 2\rangle} \wedge f) \wedge \mathrm{E}_{\langle 1\sim 3\rangle}\mathrm{X}(\neg p^{\langle 1\sim 3\rangle} \wedge r))
\end{aligned}
$$

Given a finite set of indices S and a proposition $p \in \mathrm{var}(\varphi)$, we define the following syntactic transformation:

- $\varphi[p^i/\psi]_{i \in S}$ is the formula obtained by simultaneously replacing each occurrence of p^i in φ with ψ for all $i \in S$.

Let **Rem**(ψ) be the operator that strips all labels from ψ, specifically those introduced by the **Dis-EX** rule and the $\varphi_{\mathcal{H},p}$. Consequently, Algorithm 1 correctly computes the result of forgetting atoms from a CTL(X) formula.

Algorithm 1 is space-efficient in terms of polynomial working memory: by enumerating and emitting disjuncts one by one, the algorithm limits its auxiliary space consumption to a polynomial of the input size, despite the exponential time and output scale.

Lemma 4. *Let V be a signature, $\varphi \in CTL(\mathrm{X})$, and $\psi =$AL-Forget(φ, V). Then $\psi \equiv forget(\varphi, V)$.*

The following example illustrates how the AL-Forget function (defined in Algorithm 1) operates on a concrete formula.

Example 6. *Given $\varphi = \mathrm{EX}(p \wedge f) \wedge \mathrm{EX}(\neg p \wedge r)$, we can compute forget$(\varphi, \{p\})$ as follows. Apparently,*

$$\varphi_{\mathcal{H},p} = \mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(p^{\langle 1\sim 1\rangle} \wedge f) \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(\neg p^{\langle 1\sim 2\rangle} \wedge r);$$
$$E = \{\langle 0 \sim 0\rangle, \langle 1 \sim 1\rangle, \langle 1 \sim 2\rangle\}.$$

In fact, only the labels $\langle 1 \sim 1\rangle$ and $\langle 1 \sim 2\rangle$ appear in $\varphi_{\mathcal{H},p}$. Therefore, we consider the set $E' = \{\langle 1 \sim 1\rangle, \langle 1 \sim 2\rangle\}$.

$$\begin{aligned}
\psi = &\bigvee_{S \subseteq E} \varphi_{\mathcal{H},p}[p^i/\top]_{i\in S}[p^i/\bot]_{i\in E\setminus S}\\
\equiv &\bigvee_{S \subseteq E'} \varphi_{\mathcal{H},p}[p^i/\top]_{i\in S}[p^i/\bot]_{i\in E'\setminus S}\\
\equiv &(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(\bot \wedge f) \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(\neg\bot \wedge r))\vee\\
&(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(\top \wedge f) \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(\neg\bot \wedge r))\vee\\
&(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(\bot \wedge f) \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(\neg\top \wedge r))\vee\\
&(\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}(\top \wedge f) \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}(\neg\top \wedge r))\\
\equiv &\mathrm{E}_{\langle 1\sim 1\rangle}\mathrm{X}f \wedge \mathrm{E}_{\langle 1\sim 2\rangle}\mathrm{X}r.
\end{aligned}$$

*Therefore, we have: forget$(\varphi, \{p\}) \equiv$ **Rem**$(\psi) \equiv \mathrm{EX}f \wedge \mathrm{EX}r$.*

For any signature V and formula $\varphi \in$ CTL(X), the result $\psi =$AL-Forget(φ, V) remains within CTL(X). This constructive closure, combined with the semantic equivalence established in Lemma 4, leads directly to the following result:

Theorem 2. *CTL(X) has the UI property.*

5 Uniform Interpolation Closure of CTL Fragments

This section investigates the UI closure of various CTL fragments, specifically CTL$(\mathrm{F}^<)$, CTL$(\mathrm{F}^<, \mathrm{X})$, and CTL$(\mathrm{U})$. Intuitively, the UI closure of a language $\mathcal{L}$ is its minimum extension that restores the UI property. Formally:

Definition 8 (Uniform Interpolation Closure)**.** *Let $\mathcal{L}_1$ be a temporal language. Its uniform interpolation closure, denoted by $\mathcal{L}_2$, is the unique language satisfying:*

- *$\mathcal{L}_1 \subseteq \mathcal{L}_2$;*
- *$\mathcal{L}_2$ possesses the UI property;*
- *$\mathcal{L}_2$ is minimal: for any temporal language $\mathcal{L}_3$, if $\mathcal{L}_1 \subseteq \mathcal{L}_3$ and $\mathcal{L}_3$ has the UI property, then $\mathcal{L}_2 \subseteq \mathcal{L}_3$.*

5.1 Uniform Interpolation Closure of CTL$(\mathrm{F}^<, \mathrm{X})$

Recall that CTL$(\mathrm{F}^<, \mathrm{X})$ lacks explicit definability (the *Beth property*). Consequently, we obtain the following result.

Theorem 3. *EBQCTL$(\mathrm{F}^<, \mathrm{X})$ is the UI closure of CTL$(\mathrm{F}^<, \mathrm{X})$.*

Proof. According to Proposition 4 of [Feng *et al.*, 2022], CTL$(\mathrm{F}^<, \mathrm{X})$ fails to possess the UI property. This implies that there exists a formula $\varphi \in$ CTL$(\mathrm{F}^<, \mathrm{X})$ and a set P of atoms s.t. $\exists P.\varphi$ cannot be expressed by any CTL$(\mathrm{F}^<, \mathrm{X})$ formula; i.e., the second-order quantifier in $\exists P.\varphi$, which is an EBQCTL$(\mathrm{F}^<, \mathrm{X})$ formula, cannot be eliminated.

We show below that EBQCTL$(\mathrm{F}^<, \mathrm{X})$ is the minimum extension of CTL$(\mathrm{F}^<, \mathrm{X})$ that has the UI property, i.e., for every CTL$(\mathrm{F}^<, \mathrm{X}) \subseteq \mathcal{L}$ that has the UI property, we have EBQCTL$(\mathrm{F}^<, \mathrm{X}) \subseteq \mathcal{L}$.

We prove EBQCTL$(\mathrm{F}^<, \mathrm{X}) \subseteq \mathcal{L}$ by induction on the structure of the formulas of EBQCTL$(\mathrm{F}^<, \mathrm{X})$. For any signature σ and $\varphi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$, there are three cases:

(i) $\varphi \in$ CTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$. There is a formula $\varphi' \in \mathcal{L}[\sigma]$ s.t. $\varphi \equiv \varphi'$ due to CTL$(\mathrm{F}^<, \mathrm{X}) \subseteq \mathcal{L}$.

(ii) $\varphi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$ with the form $\exists P.\psi$ and $\psi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$. By inductive hypothesis, there is a formula $\psi' \in \mathcal{L}[\sigma]$ such that $\psi \equiv \psi'$, and then $\exists P.\psi'$ is in $\mathcal{L}[\sigma]$, i.e., $\exists P.\psi \in \mathcal{L}[\sigma]$, since $\mathcal{L}$ has the UI property.

(iii) $\varphi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$ with the form $\forall P.\psi$ and $\psi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$. There is a formula $\psi' \in \mathcal{L}[\sigma]$ such that $\psi \equiv \psi'$ by inductive hypothesis, and then $\neg\psi' \in \mathcal{L}[\sigma]$ (i.e., $\neg\psi \in \mathcal{L}[\sigma]$) by the negation property. Thanks to $\forall P.\psi \equiv \neg\exists P.\neg\psi$, we have $\exists P.\neg\psi \in \mathcal{L}[\sigma]$ since $\mathcal{L}$ has the UI property, and hence $\forall P.\psi \in \mathcal{L}[\sigma]$ by the negation property.

In a word, for each formula $\varphi \in$ EBQCTL$(\mathrm{F}^<, \mathrm{X})[\sigma]$, there is a formula φ' in $\mathcal{L}[\sigma]$ s.t. $\varphi \equiv \varphi'$.

Therefore, EBQCTL$(\mathrm{F}^<, \mathrm{X})$ is the UI closure of CTL$(\mathrm{F}^<, \mathrm{X})$ since EBQCTL$(\mathrm{F}^<, \mathrm{X})$ has the UI property according to Theorem 1. □

Based on Lemma 2 and Theorem 3, the UI closure of CTL$(\mathrm{F}^<, \mathrm{X})$ is BQCTL. This follows from the fact that EBQCTL$(\mathrm{F}^<, \mathrm{X})$ is expressively equivalent to EBQCTL (and thus the μ-calculus) according to Lemma 3 and Lemma 4 of [Hossain and Laroussinie, 2021]. Furthermore, given the inclusion chain CTL$(\mathrm{F}^<, \mathrm{X}) \subseteq$ CTL $\subseteq$ BQCTL, we conclude that BQCTL also serves as the UI closure for the full CTL.

Corollary 1. *BQCTL is the UI closure of both CTL and CTL$(\mathrm{F}^<, \mathrm{X})$.*

5.2 Uniform Interpolation Closures of CTL$(\mathrm{F}^<)$ and CTL(U)

To characterize the UI closure properties of CTL$(\mathrm{F}^<)$ and CTL(U), we introduce a concept that bridges projective class and interpolation.

Definition 9 (Δ-interpolation property [Gheerbrant and ten Cate, 2009])**.** *A temporal language $\mathcal{L}$ is said to possess the Δ-interpolation property if the following holds: for any signature σ and $K \subseteq Str[\sigma]$, if both K and its complement $\overline{K}$ are projective classes of $\mathcal{L}$, then there exists an $\mathcal{L}$-formula φ such that $K = \mathsf{Mod}(\varphi)$.*

A temporal language $\mathcal{L}$ has the Δ-interpolation property if and only if for any propositional signature σ, if both a class K of initial σ-structures and its complement can be indirectly defined in $\mathcal{L}$ (as projective classes), then K must be directly definable by some formula in $\mathcal{L}[\sigma]$.

Recall that, given a signature σ, if $K \subseteq Str[\sigma]$ and $\overline{K}$ are projective classes of $\mathcal{L}$, there exist formulas $\varphi_1 \in \mathcal{L}[\tau]$ and $\varphi_2 \in \mathcal{L}[\tau']$ (where $\sigma \subseteq \tau, \tau'$) such that $K = \mathsf{Mod}(\varphi_1) \upharpoonright \sigma$ and $\overline{K} = \mathsf{Mod}(\varphi_2) \upharpoonright \sigma$. One can easily check that $\varphi_1 \models \neg\varphi_2$. Assuming that $\tau \cap \tau' = \sigma$ without loss of generality, the existence of Craig interpolation in $\mathcal{L}$ guarantees an $\alpha \in \mathcal{L}[\sigma]$ such that $\varphi_1 \models \alpha$ and $\alpha \models \neg\varphi_2$. And we can show that the formula α defines K, that is, $K = \mathsf{Mod}(\alpha)$. This result can be formally written as follows.

Lemma 5. *Let $\mathcal{L}$ be a temporal language with Craig interpolation. Then $\mathcal{L}$ has the Δ-interpolation property.*

This means that for any language $\mathcal{L}$ that has Craig interpolation, one can find a formula φ in $\mathcal{L}$ to define K whenever both K and $\overline{K}$ are projective classes of $\mathcal{L}$.

Lemma 6. *Let $\mathcal{L}$ be an extension of CTL($\mathrm{F}^<$) with Craig interpolation. Then there is $\zeta \in \mathcal{L}[\{p, q\}]$ such that $\mathsf{Mod}(\zeta) = \mathsf{Mod}(\mathrm{A}(p\mathrm{U}q))$.*

Proof. (Sketch) This can be proved by constructing two projective classes of CTL($F^<$) (also that of $\mathcal{L}$ by Lemma 1) such that an initial $\{p, q\}$-structure $\mathcal{M}$ belongs to the first class if and only if $\mathcal{M} \models \mathrm{A}(p\mathrm{U}q)$ and it belongs to the second class in all other cases. It is evident that these two projective classes are complementary to each other. By Δ-interpolation for $\mathcal{L}$ (Lemma 5), it follows that there exists $\zeta \in \mathcal{L}[\{p, q\}]$ such that $\mathsf{Mod}(\zeta) = \mathsf{Mod}(\mathrm{A}(p\mathrm{U}q))$. $\square$

Intuitively, for an extension $\mathcal{L}$ of CTL($\mathrm{F}^<$) that has the Craig interpolation property and for a given formula $\mathrm{A}(p\mathrm{U}q)$, one can identify a formula $\varphi \in \mathcal{L}[\{p, q\}]$ such that $\varphi \equiv \mathrm{A}(p\mathrm{U}q)$, and it follows that $\mathcal{L}$ is an extension of CTL(U).

Theorem 4. *Every extension of CTL($\mathrm{F}^<$) with Craig interpolation is an extension of CTL(U).*

Proof. (Sketch) Let $\mathcal{L}$ be an extension of CTL($\mathrm{F}^<$) with Craig interpolation and σ be a signature. We show by induction on the complexity of φ (i.e., the number of Boolean and temporal operators in φ) that for every $\varphi \in$ CTL(U)$[\sigma]$, there exists $\varphi' \in \mathcal{L}[\sigma]$ such that $\mathsf{Mod}(\varphi) = \mathsf{Mod}(\varphi')$.

The base case and the cases of Boolean connectives are clear.

For $\varphi = \mathrm{A}(\varphi_1\mathrm{U}\varphi_2)$, by the inductive hypothesis there exists $\varphi_1', \varphi_2' \in \mathcal{L}[\sigma]$ such that $\mathsf{Mod}(\varphi_1) = \mathsf{Mod}(\varphi_1')$ and $\mathsf{Mod}(\varphi_2) = \mathsf{Mod}(\varphi_2')$. Pick any $p, q \notin \sigma$. By Lemma 6 and the expansion property, we have:

1. There is $\zeta \in \mathcal{L}[\sigma \cup \{p, q\}]$ such that $\mathsf{Mod}(\mathrm{A}(p\mathrm{U}q)) = \mathsf{Mod}(\zeta)$.

We define φ' as $\zeta[p/\varphi_1'][q/\varphi_2'] \in \mathcal{L}[\sigma]$ (by closure under uniform substitution of $\mathcal{L}$, such a formula exists). We need to show that $\mathsf{Mod}(\mathrm{A}(\varphi_1\mathrm{U}\varphi_2)) = \mathsf{Mod}(\zeta[p/\varphi_1'][q/\varphi_2'])$. From 1 we can derive as a particular case:

2. For any $(\mathcal{T}, L, w) \in Str[\sigma \cup \{p\}]$ with $\mathcal{T} = (S, R)$, $(\mathcal{T}, L, w) \models \zeta$ iff for every path $\pi = w_0w_1\ldots$ of $\mathcal{T}$, with $w_0 = w$, there exists $k \geq 0$ such that for all $0 \leq i < k$, $(\mathcal{T}, L, w_i) \models p$ and $(\mathcal{T}, L, w_k) \models q$.

Now, by closure under uniform substitution of $\mathcal{L}$, we have:

3. For any $(\mathcal{T}, L, w) \in Str[\sigma]$ with $\mathcal{T} = (S, R)$, $(\mathcal{T}, L, w) \models \zeta[p/\varphi_1'][q/\varphi_2']$ iff $(\mathcal{T}, L, w) \models \mathrm{A}(\varphi_1\mathrm{U}\varphi_2)$. $\square$

For any formula φ in CTL(U) (resp. in CTL($\mathrm{F}^<$)) and signature V, it can be established that $\exists V.\varphi$ serves as the uniform interpolant of φ with respect to var(ϕ) $\setminus V$. Since $\exists V.\varphi$ belongs to EBQCTL(U) (resp. in EBQCTL($\mathrm{F}^<$)), following a similar argument to Theorem 3, we can prove that the UI closure of CTL(U) (resp. CTL($\mathrm{F}^<$)) is precisely EBQCTL(U) (resp. EBQCTL($\mathrm{F}^<$)).

Proposition 3. *The UI closure of CTL(U) (resp. CTL($\mathrm{F}^<$)) is EBQCTL(U) (resp. EBQCTL($\mathrm{F}^<$)).*

On the one hand, CTL($\mathrm{F}^<$) fails to possess the Craig interpolation property by Theorem 4: assuming otherwise would make CTL($\mathrm{F}^<$) an extension of CTL(U), yet U is even inexpressible in CTL($\mathrm{F}^<, \mathrm{X}$) (or CTL(F, X)). On the other hand, EBQCTL($\mathrm{F}^<$) has the UI property by Proposition 3. Consequently, CTL($\mathrm{F}^<$) is a proper fragment of EBQCTL($\mathrm{F}^<$).

In contrast to the results for CTL($\mathrm{F}^<, \mathrm{X}$) and CTL($\mathrm{F}^<$), it remains unclear whether the expressive power of CTL(U) is strictly weaker than that of EBQCTL(U).

6 Concluding Remarks

Summary. This paper presented a systematic investigation into the uniform interpolation (UI) closure properties of Computation Tree Logic (CTL) fragments. By exploring the intrinsic relationships among projective classes, forgetting, and UI, we established a series of theoretical results regarding the expressive boundaries of temporal logic:

- Based on an analysis of projective classes, we demonstrated that every extension of CTL($\mathrm{F}^<$) that possesses the Craig interpolation property is necessarily an extension of CTL(U). This finding reveals a fundamental hierarchy in how these fragments must expand to regain interpolants.
- We further proved that the UI closure of the fragment CTL($\mathrm{F}^<, \mathrm{X}$) is equivalent to EBQCTL. EBQCTL is defined as the bisimulation-invariant fragment of quantified CTL (QCTL) in prenex normal form.
- Beyond proving the absence of Craig interpolation in CTL($\mathrm{F}^<$), we characterize EBQCTL($\mathrm{F}^<$) as its minimal extension that restores the uniform interpolation property.

These conclusions identify the precise transition points where propositional quantification becomes essential to achieve structural closure under forgetting. Our work provides a robust theoretical foundation for modularity and knowledge extraction within temporal knowledge bases.

Future work. Future research includes the computational complexity of forgetting-related reasoning problems across diverse CTL fragments and forgetting-based model updating within these fragments.

Acknowledgments

We thank reviewers for their helpful comments. Renyan Feng is funded by the Guizhou University of Finance and Economics Talent Introduction Scientific Research Startup

Fund (2023YJ11) and the Science and Technology Program of Guizhou Province (Grant No. QKHZC[2023]372). Renyan Feng and Mingsen Deng are supported by the Central Guidance Fund for Local Science and Technology Development (Grant No. QKHZYD[2025]035), the Guizhou Provincial Major Scientific and Technological Program (Grant No. QKHCG[2024]ZD018), the Guizhou Provincial Key Technology R&D Program (QKBPT-GHJD [2025]007) and the Construction Project for Guizhou Provincial Key Laboratories (QKHPT-ZSYS[2025]005). Yisong's work is partially supported by the National Natural Science Foundation of China under grant 62376066 and Qiankehe Platform CXP-TXM (2025) 020.

References

[Afshari *et al.*, 2021] Bahareh Afshari, Graham E Leigh, and Guillermo Menéndez Turata. Uniform interpolation from cyclic proofs: the case of modal μ-calculus. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 335–353. Springer, 2021.

[Beth, 1953] E. W Beth. On padoa's method in the theory of definition. nederl. akad. wetensch. proc. ser. a. 56. *Nederl. Akad. Wetensch. Proc. Ser. A. 56 = Indagationes Math.*, pages 330–339, 1953.

[Bradfield and Stirling, 2007] Julian Bradfield and Colin Stirling. 12 modal μ-calculi. *Studies in logic and practical reasoning*, 3:721–756, 2007.

[Bradfield and Walukiewicz, 2018] Julian C. Bradfield and Igor Walukiewicz. The μ-calculus and model checking. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 871–919. Springer, 2018.

[Calvanese *et al.*, 2021] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Model completeness, uniform interpolants and superposition calculus: (with applications to verification of data-aware processes). *Journal of Automated Reasoning*, 65(7):941–969, 2021.

[Calvanese *et al.*, 2022] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Combination of uniform interpolants via beth definability. *Journal of Automated Reasoning*, 66(3):409–435, 2022.

[Clarke *et al.*, 1986] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[Colcombet *et al.*, 2024] Thomas Colcombet, Amina Doumane, and Denis Kuperberg. Tree algebras and bisimulation-invariant mso on finite graphs. *arXiv preprint arXiv:2407.12677*, 2024.

[D'Agostino and Hollenberg, 1996] Giovanna D'Agostino and Marco Hollenberg. Uniform interpolation, automata and the modal μ-calculus. In Marcus Kracht, Maarten de Rijke, Heinrich Wansing, and Michael Zakharyaschev, editors, *Advances in Modal Logic 1, papers from the first workshop on "Advances in Modal logic," held in Berlin, Germany, 8-10 October 1996*, pages 73–84. CSLI Publications, 1996.

[D'Agostino and Hollenberg, 2000] Giovanna D'Agostino and Marco Hollenberg. Logical questions concerning the μ-calculus: Interpolation, lyndon and łoś-tarski. *The Journal of Symbolic Logic*, 65(1):310–332, 2000.

[D'Agostino and Lenzi, 2006] Giovanna D'Agostino and Giacomo Lenzi. On modal μ-calculus with explicit interpolants. *Journal of applied logic*, 4(3):256–278, 2006.

[D'Agostino, 2008] Giovanna D'Agostino. Interpolation in non-classical logics. *Synthese*, 164(3):421–435, 2008.

[Doherty and Szalas, 2024] Patrick Doherty and Andrzej Szalas. Dual forgetting operators in the context of weakest sufficient and strongest necessary conditions. *Artif. Intell.*, 326:104036, 2024.

[Doherty *et al.*, 2001] Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas. Computing strongest necessary and weakest sufficient conditions of first-order formulas. In Bernhard Nebel, editor, *IJCAI*, pages 145–154. Morgan Kaufmann, 2001.

[Eiter and Kern-Isberner, 2019] Thomas Eiter and Gabriele Kern-Isberner. A brief survey on forgetting from a knowledge representation and reasoning perspective. *KI-Künstliche Intelligenz*, 33(1):9–33, 2019.

[Eiter and Wang, 2008] Thomas Eiter and Kewen Wang. *Semantic forgetting in answer set programming*. Elsevier Science Publishers Ltd., 2008.

[Fang *et al.*, 2019] Liangda Fang, Yongmei Liu, and Hans Van Ditmarsch. Forgetting in multi-agent modal logics. *Artificial Intelligence*, 266:51–80, 2019.

[Feng *et al.*, 2020] Renyan Feng, Erman Acar, Stefan Schlobach, Yisong Wang, and Wanwei Liu. On sufficient and necessary conditions in bounded CTL: A forgetting approach. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 361–370, 2020.

[Feng *et al.*, 2022] Renyan Feng, Erman Acar, Yisong Wang, Wanwei Liu, Stefan Schlobach, and Weiping Ding. Computing sufficient and necessary conditions in CTL: A forgetting approach. *Information Sciences*, 616:474–504, 2022.

[Feng *et al.*, 2023] Renyan Feng, Yisong Wang, Ren Qian, Lei Yang, and Panfeng Chen. Knowledge forgetting in propositional μ-calculus. *Annals of Mathematics and Artificial Intelligence*, 91(1):1–43, 2023.

[Férée *et al.*, 2024] Hugo Férée, Iris van der Giessen, Sam van Gool, and Ian Shillito. Mechanised uniform interpolation for modal logics k, gl, and isl. In Christoph Benzmüller, Marijn J.H. Heule, and Renate A. Schmidt, editors, *Automated Reasoning*, pages 43–60, Cham, 2024. Springer Nature Switzerland.

[Fussner and Santschi, 2024] Wesley Fussner and Simon Santschi. Interpolation in linear logic and related systems.

ACM Transactions on Computational Logic, 25(4), October 2024.

[Gheerbrant and ten Cate, 2009] Amélie Gheerbrant and Balder ten Cate. Craig interpolation for linear temporal languages. In *Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings*, pages 287–301, 2009.

[Ghilardi and Zawadowski, 2013] Silvio Ghilardi and Marek Zawadowski. *Sheaves, games, and model completions: A categorical approach to nonclassical propositional logics*, volume 14. Springer Science & Business Media, 2013.

[Gianola, 2023] Alessandro Gianola. Uniform interpolation for database theories. In *Verification of Data-Aware Processes via Satisfiability Modulo Theories*, pages 129–153. Springer, 2023.

[Gonçalves *et al.*, 2023] Ricardo Gonçalves, Matthias Knorr, and Joao Leite. Forgetting in answer set programming–a survey. *Theory and Practice of Logic Programming*, 23(1):111–156, 2023.

[Herzig and Mengin, 2008] Andreas Herzig and Jérôme Mengin. Uniform interpolation by resolution in modal logic. In *European Workshop on Logics in Artificial Intelligence*, pages 219–231. Springer, 2008.

[Hossain and Laroussinie, 2021] Akash Hossain and François Laroussinie. QCTL model-checking with QBF solvers. *Information and computation*, 280:104642, 2021.

[Konev *et al.*, 2008] Boris Konev, Dirk Walther, and Frank Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *International Workshop on Description Logics*, 2008.

[Kurucz *et al.*, 2023] Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. Definitions and (uniform) interpolants in first-order modal logic. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 417–428, 2023.

[Laroussinie and Markey, 2014] François Laroussinie and Nicolas Markey. Quantified CTL: expressiveness and complexity. *Logical Methods in Computer Science*, 10, 2014.

[Liberatore, 2024] Paolo Liberatore. The ghosts of forgotten things: A study on size after forgetting. *Annals of Pure and Applied Logic*, 175(8):103456, 2024.

[Lin and Reiter, 1994] Fangzhen Lin and Ray Reiter. Forget it! In *In Proceedings of the AAAI Fall Symposium on Relevance*, pages 154–159, New Orleans, US, 1994.

[Lin and Reiter, 1997] Fangzhen Lin and Raymond Reiter. How to progress a database. *Artif. Intell.*, 92(1-2):131–167, 1997.

[Liu and Claßen, 2025] Daxin Liu and Jens Claßen. On action theories with iterable first-order progression. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 15041–15048. AAAI Press, 2025.

[Lutz and Wolter, 2011] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 989–995, Barcelona, Catalonia, Spain, 2011. IJCAI/AAAI.

[Maksimova, 1991] Larisa Maksimova. Temporal logics of "the next" do not have the beth property. *Journal of Applied Non-Classical Logics*, 1:73–76, 1991.

[Marquis, 2003] Pierre Marquis. *Propositional independence: formula-variable independence and forgetting*. AI Access Foundation, 2003.

[Rohde, 2002] Philipp Rohde. Expressive power of monadic second-order logic and modal μ-calculus. *Automata Logics, and Infinite Games: A Guide to Current Research*, pages 239–257, 2002.

[Sauerwald *et al.*, 2024] Kai Sauerwald, Christoph Beierle, and Gabriele Kern-Isberner. Propositional variable forgetting and marginalization: Semantically, two sides of the same coin. In Arne Meier and Magdalena Ortiz, editors, *Foundations of Information and Knowledge Systems - 13th International Symposium, FoIKS 2024, Sheffield, UK, April 8-11, 2024, Proceedings*, volume 14589 of *Lecture Notes in Computer Science*, pages 144–162. Springer, 2024.

[Su *et al.*, 2009] Kaile Su, Abdul Sattar, Guanfeng Lv, and Yan Zhang. Variable forgetting in reasoning about knowledge. *Journal of Artificial Intelligence Research*, 35:677–716, 2009.

[Toluhi, 2023] David Toluhi. *Knowledge sharing among agents via uniform interpolation*. The University of Manchester (United Kingdom), 2023.

[Wang *et al.*, 2010] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. Forgetting for knowledge bases in DL-Lite. *Annuals of Mathematics and Artificial Intelligence*, 58(1-2):117–151, 2010.

[Wang *et al.*, 2014] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang. Knowledge forgetting in answer set programming. *Journal of Artificial Intelligence Research*, 50:31–70, 2014.

[Wolper, 1985] Pierre Wolper. The tableau method for temporal logic: An overview. *Logique Et Analyse*, 28(110):119–136, 1985.

[Zhang and Foo, 2006] Yan Zhang and Norman Y. Foo. Solving logic program conflict through strong and weak forgettings. *Artificial Intelligence*, 170(8-9):739–778, 2006.

[Zhao *et al.*, 2020] Yizheng Zhao, Renate A. Schmidt, Yuejie Wang, Xuanming Zhang, and Hao Feng. A practical approach to forgetting in description logics with nominals. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 3073–3079. AAAI Press, 2020.

[Zhou and Zhang, 2011] Yi Zhou and Yan Zhang. Bounded forgetting. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, pages 280–285, San Francisco, California, USA, 2011. AAAI Press.