

Forgetting in CTL to Compute Necessary and Sufficient Conditions

Abstract

Computation Tree Logic (CTL) is one of the central formalisms in formal verification. As a specification language, it is used to express a property that the system at hand is expected to satisfy. From both the verification and the system design points of view, some information content of such property might become irrelevant for the system due to various reasons e.g., it might become obsolete by time, or perhaps infeasible due to practical difficulties. Then, the problem arises on how to subtract such piece of information without altering the relevant system behaviour or violating the existing specifications.

To address such a scenario in a principled way, we introduce a *forgetting*-based approach in CTL and show that it can be used to compute SNC and WSC of a property under a given model. We study its theoretical properties and also show that our notion of forgetting satisfies existing essential postulates. Furthermore, we analyse the computational complexity of basic tasks, including various results for the relevant fragment CTL_{AF} .

1 Introduction

Consider a car-manufacturing company which produces two types of automobiles: a sedan car (basically a four-doored classical passenger car) and a sports car. No matter a sedan or a sports car, both production lines are subject to a single standard criterion which is indispensable: safety restrictions. Complementing such a shared feature, there are also differences aligned with these types in general. That is, a sedan car is produced with a small engine, while a sports car is produced with a large one. Moreover, due to its large amount of production, the sedan car is subject to some very restrictive low-carbon emission regulations, while a sports car is not. In the verge of shifting to an upcoming new engine technology, the company aims to adapt the sedan production to electrical engines. Such major shift in production also comes with one in regulations; Electric sedans are not subject to low-carbon emission restrictions any more. In fact, due to the difference in its underlying technology, a sedan car drastically emit much less carbon, hence such standard is obsolete, and

can be dropped. Yet, dropping some restrictions in a large and complex production system (like the one in automotive industry) without affecting the working system components or violating dependent specifications is a non-trivial task.

Similar situations arise in many different domains such as business-process modelling, financial systems, software development, concurrent systems and more [Baier and Katoen, 2008]. In general, some information content of such property might become irrelevant for the system due to various reasons e.g., it might become obsolete by time like in the above example, or perhaps becomes infeasible due to practical difficulties. Then, the problem arises on how to subtract such piece of information without altering the behaviour of the relevant system components or violating the existing specifications. Moreover, in such a scenario, two logical notions introduced by E. Dijkstra in [Dijkstra, 1978] are very informative: the *strongest necessary condition* (SNC) and the *weakest sufficient condition* (WSC) of a given specification. These correspond to *most general consequence* and the *most specific abduction* of such specification, respectively.

To address such a scenario in a principled way, we employ a method based on formal verification which is useful for abstracting away the domain-specific problems, and rather focusing on conceptual issues. In particular, we introduce a *forgetting* based approach in Computation Tree Logic CTL a central formalism in formal verification, and show that it can be. WSC was widely used in program *verification*, especially in generating counterexamples [Daijler *et al.*, 2018] and refinement of system [Woodcock and Morgan, 1990]. *Computation Tree Logic* (CTL) [Clarke and Emerson, 1981] Model modification, which has been developed in [Ramdani *et al.*, 2019; Martinez-Araiza and López-Mellado, 2016; Ding and Zhang, 2006], is an extension of refinement of system. This paper explore a method to compute the WSC of a property (a CTL formula) under a given model system that may be modified for guiding CTL Model modification. It is known that the computing of WSC for code fragment S with respect to assertion Q requires S must terminate [Tremblay, 1996] since to it just concerns relation among input values and output values. However, in the case of model checking, it concerns properties about execution runs, which may not terminate. It can be shown by the following example.

Computation Tree Logic (CTL) [Clarke and Emerson, 1981] is one of the central formalisms in formal verification.

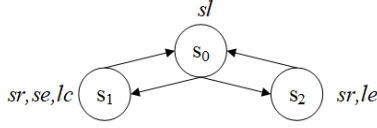


Figure 1: Car Engine Manufacturing Scenario

As a specification language, it is used to express a property that the system at hand is expected to satisfy. From both the verification and the system design points of view, some information content of such property might become irrelevant due to various reasons e.g., it might become obsolete by time, or perhaps infeasible due to practical difficulties.

To address such a scenario in a principled way, we introduce a *forgetting*-based approach in CTL and show that it can be used to compute SNC and WSC of a property under a given model. We study its theoretical properties and also show that our notion of forgetting satisfies existing essential postulates. Furthermore, we analyse the computational complexity of basic tasks, including various results for the relevant fragment CTL_{AF} .

Example 1 The scenario we mentioned concerning car-engine manufacturing can be represented by the following Kripke structure $\mathcal{M} = (S, R, L, s_0)$ in Figure 1 on $V = \{sl, sr, se, le, lc\}$ whose elements correspond to *select*, *safety restrictions*, *small engine*, *large engine* and *low-carbon emission requirements*, respectively. Moreover, s_0 is the initial state where we select either choose producing an engine for *sedan* which corresponds to the state s_1 or a *sports car* which corresponds to the state s_2 . Since only a single-type of production is possible at a time, after each production state (s_1 or s_2), we turn back to the initial state (s_0) to start over. Let $\varphi = \text{AGAF}(sr)$, which means sr will be satisfied infinitely many times in the structure, be a CTL formula.

We can decide $(\mathcal{M}, s_0) \not\models \varphi$ easily since this structure does not contain the atom r . In order for (\mathcal{M}, s_0) to satisfy φ , we should find a condition ψ such that $(\mathcal{M}, s_0) \models \psi \supset \varphi$. As we know that if this condition exists, there are also many other conditions that satisfy the need. In this is the case, we can analyze in advance the set of possible atomic propositions that make up the condition. Then we can find this condition in the set only. And the smaller the set, the easier to find out the condition. In this paper, we always assume that the condition is a property defined on the specified atomic proposition set V , for our example $V = \{p, r\}$, and find the weakest property (that is, the weakest sufficient condition) satisfying the condition on the set. Finding this property is called discovering theorem by Lin in [Lin, 2018]. Inspired by the forgetting-based method to compute SNC (WSC) [Lin, 2001], in this paper, we tackle this problem by proposing a semantic forgetting for CTL.

However, since \mathcal{M} is a Kripke structure, it needs to be converted into a logical formula (theory), that is the characterizing formula, which is in CTL proposed in [Browne et al., 1988]. We find the WSC in a set V of atoms, hence a set-based bisimulation between two K-structures (a Kripke structure with a state in it), V -bisimulation, and characterizing formula on V will be proposed in this paper. Our

V -bisimulation is a more general bisimulation than others. On the one hand, the above set-based bisimulation is an extension of the bisimulation-equivalence of Definition 7.1 in [Baier and Katoen, 2008] in the sense that, if $V = \mathcal{A}$, then our bisimulation is almost same to the latter. On the other hand, the above set-based bisimulation notion is similar to the state equivalence in [Browne et al., 1988]. But it is different in the sense that ours is defined on K-structures, while it is defined on states in [Browne et al., 1988]. Moreover, the set-based bisimulation notion is also different from the state-based bisimulation notion of Definition 7.7 in [Baier and Katoen, 2008], which is defined for states of a given K-structure.

As a logical notion, *forgetting* was first formally defined in propositional and first order logics by Lin and Reiter [Lin and Reiter, 1994]. Over the last twenty years, researchers have developed forgetting notions and theories not only in classical logic but also in other non-classical logic systems [Eiter and Kern-Isberner, 2019], such as forgetting in logic programs under answer set/stable model semantics [Zhang and Foo, 2006; Eiter and Wang, 2008; Wong, 2009; Wang et al., 2012; Wang et al., 2013], forgetting in description logic [Wang et al., 2010; Lutz and Wolter, 2011; Zhao and Schmidt, 2017] and knowledge forgetting in modal logic [Zhang and Zhou, 2009; Su et al., 2009; Liu and Wen, 2011; Fang et al., 2019]. In application, forgetting has been used in planning [Lin, 2003], conflict solving [Lang and Marquis, 2010; Zhang et al., 2005], creating restricted views of ontologies [Zhao and Schmidt, 2017], strongest and weakest definitions [Lang and Marquis, 2008], SNC (WSC) [Lin, 2001] and so on.

Although forgetting has been extensively investigated from various aspects of different logical systems, the existing forgetting techniques are not directly applicable in CTL. For instance, in propositional forgetting theory, forgetting atom q from φ is equivalent to a formula $\varphi[q/\top] \vee \varphi[q/\perp]$, where $\varphi[q/X]$ is a formula obtained from φ by replacing each q with X ($X \in \{\top, \perp\}$). This method cannot be extended to a CTL formula. Consider a CTL formula $\psi = \text{AG}p \wedge \neg \text{AG}q \wedge \neg \text{AG}\neg q$. If we want to forget atom q from ψ by using the above method, we would have $\psi[q/\top] \vee \psi[q/\perp] \equiv \perp$. This is obviously not correct since after forgetting q this specification should not become inconsistent. Similar to [Zhang and Zhou, 2009], we research forgetting in CTL from the semantic forgetting point of view. And it is shown that our definition of forgetting satisfies those four postulates of forgetting presented in [Zhang and Zhou, 2009].

The rest of the paper is organised as follows. Section 2 introduces the related notions for forgetting in CTL, including the syntax and semantics of CTL. A formal definition of concept forgetting and its properties for CTL follows in Section 3. Section 4 explores the relation between forgetting and SNC (WSC), along with a computational complexity analysis. From the point of view of model, we propose an algorithm for computing forgetting on CTL in Section 5 and study its complexity. Finally, we conclude this paper.

2 Preliminaries

We start with some technical and notational preliminaries. Throughout this paper, we fix a finite set \mathcal{A} of propositional variables (or atoms), and use V, V' for subsets of \mathcal{A} .

2.1 Model structure in CTL

In general, a transition system can be described by a *model structure* (or *Kripke structure*) (see [Baier and Katoen, 2008] for details). A model structure is a triple $\mathcal{M} = (S, R, L)$, where

- S is a finite nonempty set of states ¹,
- $R \subseteq S \times S$ and, for each $s \in S$, there is $s' \in S$ such that $(s, s') \in R$,
- L is a labeling function $S \rightarrow 2^{\mathcal{A}}$.

We call a model structure \mathcal{M} on a set V of atoms if $L : S \rightarrow 2^V$, i.e., the labelling function L map every state to V (not the \mathcal{A}). A *path* π_{s_i} start from s_i of \mathcal{M} is an infinite sequence of states $\pi_{s_i} = (s_i, s_{i+1}, s_{i+2}, \dots)$, where for each j ($0 \leq i \leq j$), $(s_j, s_{j+1}) \in R$. By $s' \in \pi_{s_i}$ we mean that s' is a state in the path π_{s_i} . A state $s \in S$ is *initial* if for any state $s' \in S$, there is a path π_s s.t $s' \in \pi_s$. We denote this model structure as (S, R, L, s_0) , where s_0 is initial.

For a given model structure (S, R, L, s_0) and $s \in S$, the *computation tree* $\text{Tr}_n^{\mathcal{M}}(s)$ of \mathcal{M} (or simply $\text{Tr}_n(s)$), that has depth n and is rooted at s , is recursively defined as [Browne et al., 1988], for $n \geq 0$,

- $\text{Tr}_0(s)$ consists of a single node s with label s .
- $\text{Tr}_{n+1}(s)$ has as its root a node m with label s , and if $(s, s') \in R$ then the node m has a subtree $\text{Tr}_n(s')$.

By s_n we mean a n th level node of tree $\text{Tr}_m(s)$ ($m \geq n$).

A *K-structure* (or *K-interpretation*) is a model structure $\mathcal{M} = (S, R, L, s_0)$ associating with a state $s \in S$, which is written as (\mathcal{M}, s) for convenience in the following. In the case $s = s_0$ is an initial state of \mathcal{M} , the K-structure is *initial*.

2.2 Syntax and semantics of CTL

In the following we briefly review the basic syntax and semantics of the CTL [Clarke et al., 1986]. The *signature* of the language \mathcal{L} of CTL includes:

- a finite set of Boolean variables, called *atoms* of \mathcal{L} : \mathcal{A} ;
- constant symbols: \perp and \top ;
- the classical connectives: \vee and \neg ;
- the path quantifiers: A and E ;
- the temporal operators: X, F, G, U and W , that means ‘neXt state’, ‘some Future state’, ‘all future states (Globally)’, ‘Until’ and ‘Unless’, respectively;
- parentheses: (and).

The (*existential normal form* or *ENF in short*) formulas of \mathcal{L} are inductively defined via a Backus Naur form:

$$\phi ::= \perp \mid \top \mid p \mid \neg\phi \mid \phi \vee \phi \mid \text{EX}\phi \mid \text{EG}\phi \mid E[\phi \text{ U } \phi] \quad (1)$$

¹Indeed, every state is identified by a configuration of atoms i.e., which holds in that state.

where $p \in \mathcal{A}$. The formulas $\phi \wedge \psi$ and $\phi \rightarrow \psi$ are defined in a standard manner of propositional logic. The other form formulas of \mathcal{L} are abbreviated using the forms of (1).

We are now in the position to define the semantics of \mathcal{L} . Let $\mathcal{M} = (S, R, L, s_0)$ be a model structure, $s \in S$ and ϕ a formula of \mathcal{L} . The *satisfiability* relationship between (\mathcal{M}, s) and ϕ , written $(\mathcal{M}, s) \models \phi$, is inductively defined on the structure of ϕ as follows:

- $(\mathcal{M}, s) \not\models \perp$ and $(\mathcal{M}, s) \models \top$;
- $(\mathcal{M}, s) \models p$ iff $p \in L(s)$;
- $(\mathcal{M}, s) \models \phi_1 \vee \phi_2$ iff $(\mathcal{M}, s) \models \phi_1$ or $(\mathcal{M}, s) \models \phi_2$;
- $(\mathcal{M}, s) \models \neg\phi$ iff $(\mathcal{M}, s) \not\models \phi$;
- $(\mathcal{M}, s) \models \text{EX}\phi$ iff $(\mathcal{M}, s_1) \models \phi$ for some $s_1 \in S$ and $(s, s_1) \in R$;
- $(\mathcal{M}, s) \models \text{EG}\phi$ iff \mathcal{M} has a path $(s_1 = s, s_2, \dots)$ such that $(\mathcal{M}, s_i) \models \phi$ for each $i \geq 1$;
- $(\mathcal{M}, s) \models E[\phi_1 \text{ U } \phi_2]$ iff \mathcal{M} has a path $(s_1 = s, s_2, \dots)$ such that, for some $i \geq 1$, $(\mathcal{M}, s_i) \models \phi_2$ and $(\mathcal{M}, s_j) \models \phi_1$ for each $1 \leq j < i$.

Similar to the work in [Browne et al., 1988; Bolotov, 1999], only initial K-structures are considered to be candidate models in the following, unless otherwise noted. Formally, an initial K-structure \mathcal{K} is a *model* of a formula ϕ whenever $\mathcal{K} \models \phi$. We denote $\text{Mod}(\phi)$ the set of models of ϕ . The formula ϕ is *satisfiable* if $\text{Mod}(\phi) \neq \emptyset$. Since the states in model structure is finite, $\text{Mod}(\phi)$ is finite for any formula ϕ up to equivalence.

Let ϕ_1 and ϕ_2 be two formulas. By $\phi_1 \models \phi_2$ we mean $\text{Mod}(\phi_1) \subseteq \text{Mod}(\phi_2)$, and by $\phi_1 \equiv \phi_2$, we mean $\phi_1 \models \phi_2$ and $\phi_2 \models \phi_1$. In this case ϕ_1 is *equivalent* to ϕ_2 . The set of atoms occurring in ϕ_1 , is denoted by $\text{Var}(\phi_1)$. ϕ_1 is *V-irrelevant*, written $\text{IR}(\phi_1, V)$, if there is a formula ψ with $\text{Var}(\psi) \cap V = \emptyset$ such that $\phi_1 \equiv \psi$.

3 Forgetting in CTL

In this section, we will define the forgetting in CTL by *V*-bisimulation constructed via a set-based bisimulation. Besides, some properties of forgetting are also explored. For convenience, let $\mathcal{M} = (S, R, L, s_0)$, $\mathcal{M}' = (S', R', L', s'_0)$ and $\mathcal{K}_i = (\mathcal{M}_i, s_i)$ with $\mathcal{M}_i = (S_i, R_i, L_i, s_i^0)$, $s_i \in S_i$ (is any state in S_i) and $i \in \mathbb{N}$.

3.1 Set-based bisimulation

To present a formal definition of forgetting, we need the concept of *V*-bisimulation. Inspired by the notion of bisimulation in [Browne et al., 1988], we define the relations $\mathcal{B}_0, \mathcal{B}_1, \dots$ between K-structures on V as follows: let $\mathcal{K}_i = (\mathcal{M}_i, s_i)$ with $i \in \{1, 2\}$,

- $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_0$ if $L_1(s_1) - V = L_2(s_2) - V$;
- for $n \geq 0$, $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}$ if:
 - $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_0$,
 - for every $(s_1, s'_1) \in R_1$, there is a $(s_2, s'_2) \in R_2$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}_n$, and

- for every $(s_2, s'_2) \in R_2$, there is a $(s_1, s'_1) \in R_1$ such that $(K'_1, K'_2) \in \mathcal{B}_n$,

where $K'_i = (\mathcal{M}_i, s'_i)$ with $i \in \{1, 2\}$.

Now, we define the notion of V -bisimulation between \mathbf{K} -structures:

Definition 1 (V -bisimulation) Let $V \subseteq \mathcal{A}$. Given two \mathbf{K} -structures \mathcal{K}_1 and \mathcal{K}_2 are V -bisimilar, denoted $\mathcal{K}_1 \leftrightarrow_V \mathcal{K}_2$ if and only if $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i$ for all $i \geq 0$. Moreover, two paths $\pi_i = (s_{i,1}, s_{i,2}, \dots)$ of \mathcal{M}_i with $i \in \{1, 2\}$ are V -bisimilar if $\mathcal{K}_{1,j} \leftrightarrow_V \mathcal{K}_{2,j}$ for every $j \geq 1$ where $\mathcal{K}_{i,j} = (\mathcal{M}_i, s_{i,j})$.

It is apparent that \leftrightarrow_V is a binary relation. In the sequel, we abbreviate $\mathcal{K}_1 \leftrightarrow_V \mathcal{K}_2$ by $s_1 \leftrightarrow_V s_2$ whenever the underlying model structures of states s_1 and s_2 are clear from the context.

Lemma 1 The relation \leftrightarrow_V is an equivalence relation.

Besides, we have the following properties:

Proposition 1 Let $i \in \{1, 2\}$, $V_1, V_2 \subseteq \mathcal{A}$, s'_i be two states and π'_i be two paths, and $\mathcal{K}_i = (\mathcal{M}_i, s_i)$ ($i = 1, 2, 3$) be \mathbf{K} -structures such that $\mathcal{K}_1 \leftrightarrow_{V_1} \mathcal{K}_2$ and $\mathcal{K}_2 \leftrightarrow_{V_2} \mathcal{K}_3$. Then:

- (i) $s'_1 \leftrightarrow_{V_1} s'_2$ ($i = 1, 2$) implies $s'_1 \leftrightarrow_{V_1 \cup V_2} s'_2$;
- (ii) $\pi'_1 \leftrightarrow_{V_1} \pi'_2$ ($i = 1, 2$) implies $\pi'_1 \leftrightarrow_{V_1 \cup V_2} \pi'_2$;
- (iii) for each path π_{s_1} of \mathcal{M}_1 there is a path π_{s_2} of \mathcal{M}_2 such that $\pi_{s_1} \leftrightarrow_{V_1} \pi_{s_2}$, and vice versa;
- (iv) $\mathcal{K}_1 \leftrightarrow_{V_1 \cup V_2} \mathcal{K}_3$;
- (v) If $V_1 \subseteq V_2$ then $\mathcal{K}_1 \leftrightarrow_{V_2} \mathcal{K}_2$.

Intuitively, if two \mathbf{K} -structures are V -bisimilar, then they satisfy the same formula φ that dose not contain any atoms in V , i.e. $\text{IR}(\varphi, V)$.

Theorem 1 Let $V \subseteq \mathcal{A}$, \mathcal{K}_i ($i = 1, 2$) be two \mathbf{K} -structures such that $\mathcal{K}_1 \leftrightarrow_V \mathcal{K}_2$ and ϕ a formula with $\text{IR}(\phi, V)$. Then $\mathcal{K}_1 \models \phi$ if and only if $\mathcal{K}_2 \models \phi$.

Let $V \subseteq \mathcal{A}$, \mathcal{M}_i ($i = 1, 2$) be model structures. A computation tree $\text{Tr}_n(s_1)$ of \mathcal{M}_1 is V -bisimilar to a computation tree $\text{Tr}_n(s_2)$ of \mathcal{M}_2 , written $(\mathcal{M}_1, \text{Tr}_n(s_1)) \leftrightarrow_V (\mathcal{M}_2, \text{Tr}_n(s_2))$ (or simply $\text{Tr}_n(s_1) \leftrightarrow_V \text{Tr}_n(s_2)$), if

- $L_1(s_1) - V = L_2(s_2) - V$,
- for every subtree $\text{Tr}_{n-1}(s'_1)$ of $\text{Tr}_n(s_1)$, $\text{Tr}_n(s_2)$ has a subtree $\text{Tr}_{n-1}(s'_2)$ such that $\text{Tr}_{n-1}(s'_1) \leftrightarrow_V \text{Tr}_{n-1}(s'_2)$, and vice versa.

Note that the last condition in the above definition hold trivially for $n = 0$.

Proposition 2 Let $V \subseteq \mathcal{A}$ and (\mathcal{M}_i, s_i) ($i = 1, 2$) be two \mathbf{K} -structures. Then

$(s_1, s_2) \in \mathcal{B}_n$ iff $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for every $0 \leq j \leq n$.

This means that $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for all $j \geq 0$ if $s_1 \leftrightarrow_V s_2$, otherwise there is some k such that $\text{Tr}_k(s_1)$ and $\text{Tr}_k(s_2)$ are not V -bisimilar.

Proposition 3 Let $V \subseteq \mathcal{A}$, \mathcal{M} be a model structure and $s, s' \in S$ such that $s \not\leftrightarrow_V s'$. There exists a least k such that $\text{Tr}_k(s)$ and $\text{Tr}_k(s')$ are not V -bisimilar.

In this case the model structure \mathcal{M} is called V -distinguishable (by states s and s' at the least depth k), which is denoted by $\text{dis}_V(\mathcal{M}, s, s', k)$. The V -characterization number of \mathcal{M} , written $\text{ch}(\mathcal{M}, V)$, is defined as

$$\text{ch}(\mathcal{M}, V) = \begin{cases} \max\{k \mid s, s' \in S \text{ and } \text{dis}_V(\mathcal{M}, s, s', k)\}, & \mathcal{M} \text{ is } V\text{-distinguishable}; \\ \min\{k \mid \mathcal{B}_k = \mathcal{B}_{k+1}\}, & \text{otherwise.} \end{cases}$$

3.2 Characterization of Initial \mathbf{K} -structure

In order to introduce our notion of forgetting, and to compute strongest necessary and weakest sufficient conditions, we need a formula that captures the initial \mathbf{K} -structure on V syntactically. We call such formula as characterizing formula. In the following, we present such characterization.

Given a set $V \subseteq \mathcal{A}$, we define a formula φ of V (that is $\text{Var}(\varphi) \subseteq V$) in CTL that describes a computation tree.

Definition 2 Let $V \subseteq \mathcal{A}$, $\mathcal{M} = (S, R, L, s_0)$ be a model structure and $s \in S$. The characterizing formula of the computation tree $\text{Tr}_n(s)$ on V , written $\mathcal{F}_V(\text{Tr}_n(s))$, is defined recursively as:

$$\begin{aligned} \mathcal{F}_V(\text{Tr}_0(s)) &= \bigwedge_{p \in V \cap L(s)} p \wedge \bigwedge_{q \in V - L(s)} \neg q, \\ \mathcal{F}_V(\text{Tr}_{k+1}(s)) &= \bigwedge_{(s, s') \in R} \text{EX} \mathcal{F}_V(\text{Tr}_k(s')) \\ &\quad \wedge \text{AX} \left(\bigvee_{(s, s') \in R} \mathcal{F}_V(\text{Tr}_k(s')) \right) \wedge \mathcal{F}_V(\text{Tr}_0(s)) \end{aligned}$$

for $k \geq 0$.

The characterizing formula of a computation tree formally exhibit the content of each node on V (i.e., atoms that are true at this node if they are in V , false otherwise) and the temporal relation between states recursively. The following result shows that the V -bisimulation between two computation trees imply the semantic equivalence of the corresponding characterizing formulas.

Lemma 2 Let $V \subseteq \mathcal{A}$, \mathcal{M} and \mathcal{M}' be two model structures, $s \in S$, $s' \in S'$ and $n \geq 0$. If $\text{Tr}_n(s) \leftrightarrow_V \text{Tr}_n(s')$, then $\mathcal{F}_V(\text{Tr}_n(s)) \equiv \mathcal{F}_V(\text{Tr}_n(s'))$.

Let $s' = s$, it shows that for any formula φ of V , if φ is a characterizing formula of $\text{Tr}_n(s)$ then $\varphi \equiv \mathcal{F}_V(\text{Tr}_n(s))$.

Let $V \subseteq \mathcal{A}$, $\mathcal{K} = (\mathcal{M}, s_0)$ be an initial \mathbf{K} -structure and $T(s') = \mathcal{F}_V(\text{Tr}_c(s'))$. The characterizing formula of \mathcal{K} on V , written $\mathcal{F}_V(\mathcal{M}, s_0)$ (or $\mathcal{F}_V(\mathcal{K})$ in short), is defined as the following formula:

$$\mathcal{F}_V(\text{Tr}_c(s_0)) \wedge \bigwedge_{s \in S} \text{AG} \left(\mathcal{F}_V(\text{Tr}_c(s)) \rightarrow \bigwedge_{(s, s') \in R} \text{EXT}(s') \wedge \text{AX} \bigvee_{(s, s') \in R} T(s') \right)$$

where $c = \text{ch}(\mathcal{M}, V)$. It is apparent that $\text{IR}(\mathcal{F}_V(\mathcal{M}, s_0), \bar{V})$.

The following example show how to compute characterizing formula:

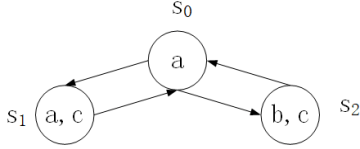


Figure 2: A simple Kripke structure

Example 2 Let $\mathcal{K} = (\mathcal{M}, s_0)$ in Figure 2 be an initial \mathcal{K} -structure and $V = \{a, b\}$. It is apparent that $\text{Tr}_0(s_0) \leftrightarrow_V \text{Tr}_0(s_1)$ because $L(s_0) - \bar{V} = L(s_1) - \bar{V}$, and $\text{Tr}_1(s_0) \not\leftrightarrow_V \text{Tr}_1(s_1)$ since there is $(s_0, s_2) \in R$ such that for any $(s_1, s') \in R$ since there is $L(s_2) - \bar{V} \neq L(s') - \bar{V}$ (as there is only one immediate successor $s' = s_0$). Hence, we have that \mathcal{M} is \bar{V} -distinguished by state s_0 and s_1 at the least depth 1, i.e. $\text{dis}_{\bar{V}}(\mathcal{M}, s_0, s_1, 1)$. Similarly, we have $\text{dis}_{\bar{V}}(\mathcal{M}, s_0, s_2, 0)$ and $\text{dis}_{\bar{V}}(\mathcal{M}, s_1, s_2, 0)$. Therefore, $\text{ch}(\mathcal{M}, \bar{V}) = \max\{k \mid s, s' \in S \text{ and } \text{dis}_{\bar{V}}(\mathcal{M}, s, s', k)\} = 1$. Then we have:

$$\begin{aligned} \mathcal{F}_V(\text{Tr}_0(s_0)) &= a \wedge \neg b, \\ \mathcal{F}_V(\text{Tr}_0(s_1)) &= a \wedge \neg b, \\ \mathcal{F}_V(\text{Tr}_0(s_2)) &= b \wedge \neg a, \\ \mathcal{F}_V(\text{Tr}_1(s_0)) &= \text{EX}(a \wedge \neg b) \wedge \text{EX}(b \wedge \neg a) \wedge \text{AX}((a \wedge \neg b) \vee (b \wedge \neg a)) \wedge (a \wedge \neg b), \\ \mathcal{F}_V(\text{Tr}_1(s_1)) &= \text{EX}(a \wedge \neg b) \wedge \text{AX}(a \wedge \neg b) \wedge (a \wedge \neg b), \\ \mathcal{F}_V(\text{Tr}_1(s_2)) &= \text{EX}(a \wedge \neg b) \wedge \text{AX}(a \wedge \neg b) \wedge (b \wedge \neg a). \end{aligned}$$

Then it is easy to obtain $\mathcal{F}_V(\mathcal{M}, s_0)$ which is the characterizing formula of \mathcal{K} on V .

By the following theorem, we also have that given $V \subseteq \mathcal{A}$, the characterizing formula of an initial \mathcal{K} -structure is unique (up to semantic equivalence) which describes this initial \mathcal{K} -structure on V .

Theorem 2 Given $V \subseteq \mathcal{A}$, let $\mathcal{M} = (S, R, L, s_0)$ and $\mathcal{M}' = (S', R', L', s'_0)$ be two model structures. Then,

- (i) $(\mathcal{M}', s'_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$ iff $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$;
- (ii) $s_0 \leftrightarrow_V s'_0$ implies $\mathcal{F}_V(\mathcal{M}, s_0) \equiv \mathcal{F}_V(\mathcal{M}', s'_0)$.

Moreover, we know that any initial \mathcal{K} -structure can be described as a CTL formula from the definition of characterizing formula. Then,

Lemma 3 Let φ be a formula. We have

$$\varphi \equiv \bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_A(\mathcal{M}, s_0). \quad (2)$$

It follows that any CTL formula can be rewritten in the form of a characterizing formula (in the form of a disjunction of characterizing formulas; each corresponds to a single model).

3.3 Semantic Properties of Forgetting in CTL

In this subsection we will give the definition of forgetting in CTL and study its semantic properties. We will first show via a representation theorem that our definition of forgetting correspond to the readily existing notion of forgetting which is characterised by several desirable properties (also called

postulates) suggested in [Zhang and Zhou, 2009]. Next, we discuss various additional semantic properties of forgetting.

Now, we give the formal definition of forgetting in CTL from the semantic point view.

Definition 3 (Forgetting) Let $V \subseteq \mathcal{A}$ and ϕ a formula. A formula ψ with $\text{Var}(\psi) \cap V = \emptyset$ is a result of forgetting V from ϕ , if

$$\text{Mod}(\psi) = \{\mathcal{K} \text{ is initial} \mid \exists \mathcal{K}' \in \text{Mod}(\phi) \ \& \ \mathcal{K}' \leftrightarrow_V \mathcal{K}\}. \quad (3)$$

Note that if both ψ and ψ' are results of forgetting V from ϕ , then $\text{Mod}(\psi) = \text{Mod}(\psi')$, i.e., ψ and ψ' have the same models. In the sense of equivalence, the forgetting result is unique (up to equivalence). By Lemma 3, such a formula always exists, which is equivalent to

$$\bigvee_{\mathcal{K} \in \{\mathcal{K}' \mid \exists \mathcal{K}'' \in \text{Mod}(\phi) \text{ and } \mathcal{K}'' \leftrightarrow_V \mathcal{K}'\}} \mathcal{F}_{\bar{V}}(\mathcal{K}).$$

For this reason, the forgetting result is denoted by $\text{F}_{\text{CTL}}(\phi, V)$.

Assume you are given a formula φ , and φ' is the formula after forgetting V , then we have the following desired properties, also called *postulates* of forgetting [Zhang and Zhou, 2009].

- **Weakening (W)**: $\varphi \models \varphi'$;
- **Positive Persistence (PP)**: Given $\eta \in \text{CTL}$ if $\text{IR}(\eta, V)$ and $\varphi \models \eta$, then $\varphi' \models \eta$;
- **Negative Persistence (NP)**: Given $\eta \in \text{CTL}$ if $\text{IR}(\eta, V)$ and $\varphi \not\models \eta$, then $\varphi' \not\models \eta$;
- **Irrelevance (IR)**: $\text{IR}(\varphi', V)$.

Intuitive enough, the postulate (W) says, forgetting weakens the original formula. (PP) and (NP) correspond to the fact that so long as forgotten atoms V are irrelevant to the remaining positive and the negative information, respectively, they do not affect such information. (IR) states that forgotten atoms V are not relevant for the final formula (i.e., φ' is V -irrelevant).

Theorem 3 (Representation Theorem) Let φ , φ' and ϕ be CTL formulas and $V \subseteq \mathcal{A}$. Then the following statements are equivalent:

- (i) $\varphi' \equiv \text{F}_{\text{CTL}}(\varphi, V)$,
- (ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi \text{ and } \text{IR}(\phi, V)\}$,
- (iii) Postulates (W), (PP), (NP) and (IR) hold.

The above theorem says that, the notion of forgetting we defined, satisfies (and entailed by) the four postulates of forgetting. Moreover, CTL is closed under our definition of forgetting, i.e., for any CTL formula the result of forgetting is also a CTL formula.

Lemma 4 Let φ and α be two CTL formulae and $q \in \text{Var}(\varphi) \cup \text{Var}(\alpha)$. Then $\text{F}_{\text{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q) \equiv \varphi$.

Proposition 4 Given a formula $\varphi \in \text{CTL}$, V a set of atoms and p an atom such that $p \notin V$. Then,

$$\text{F}_{\text{CTL}}(\varphi, \{p\} \cup V) \equiv \text{F}_{\text{CTL}}(\text{F}_{\text{CTL}}(\varphi, p), V).$$

This means that the result of forgetting V from φ can be obtained by forgetting atoms in V one by one. Moreover, the order of atoms does not matter (commutativity), which follows from Proposition 4.

Corollary 4 (Commutativity) *Let φ be a formula and $V_i \subseteq \mathcal{A}$ ($i = 1, 2$). Then:*

$$F_{CTL}(\varphi, V_1 \cup V_2) \equiv F_{CTL}(F_{CTL}(\varphi, V_1), V_2).$$

The following results, which are satisfied in both classical propositional logic and modal logic **S5** [Zhang and Zhou, 2009], further illustrate other essential semantic properties of forgetting.

Proposition 5 *Let $\varphi, \varphi_i, \psi_i$ ($i = 1, 2$) be formulas and $V \subseteq \mathcal{A}$. We have*

- (i) $F_{CTL}(\varphi, V)$ is satisfiable iff φ is;
- (ii) If $\varphi_1 \equiv \varphi_2$, then $F_{CTL}(\varphi_1, V) \equiv F_{CTL}(\varphi_2, V)$;
- (iii) If $\varphi_1 \models \varphi_2$, then $F_{CTL}(\varphi_1, V) \models F_{CTL}(\varphi_2, V)$;
- (iv) $F_{CTL}(\psi_1 \vee \psi_2, V) \equiv F_{CTL}(\psi_1, V) \vee F_{CTL}(\psi_2, V)$;
- (v) $F_{CTL}(\psi_1 \wedge \psi_2, V) \equiv F_{CTL}(\psi_1, V) \wedge F_{CTL}(\psi_2, V)$;

Another interesting result is that the forgetting of $QT\varphi$ ($Q \in \{E, A\}, T \in \{F, X\}$) on $V \subseteq \mathcal{A}$ can be computed by $QTF_{CTL}(\varphi, V)$. This gives us a convenient method to compute forgetting, since we can push the forgetting operator to a subformula without affecting the semantics, and rather compute the forgetting on this subformula.

Proposition 6 (Homogeneity) *Let $V \subseteq \mathcal{A}$ and $\phi \in CTL$,*

- (i) $F_{CTL}(AX\phi, V) \equiv AXF_{CTL}(\phi, V)$.
- (ii) $F_{CTL}(EX\phi, V) \equiv EXF_{CTL}(\phi, V)$.
- (iii) $F_{CTL}(AF\phi, V) \equiv AFF_{CTL}(\phi, V)$.
- (iv) $F_{CTL}(EF\phi, V) \equiv EFF_{CTL}(\phi, V)$.

3.4 Complexity Results

In the following, we outline the computational complexity of the various tasks regarding the forgetting in CTL and its popular fragment CTL_{AF} . It turns out that the model-checking on forgetting without any restriction is NP-complete.

Proposition 7 (Model Checking on Forgetting)

Let (\mathcal{M}, s_0) be an initial \mathcal{K} -structure, φ be a CTL formula and V a set of atoms. Deciding whether (\mathcal{M}, s_0) is a model of $F_{CTL}(\varphi, V)$ is NP-complete.

The fragment of CTL, in which each formula contains only AF temporal connective correspond to specifications that are desired to hold in all branches eventually. Such properties are of special interest in concurrent systems e.g., mutual exclusion and waiting events [Baier and Katoen, 2008]. In the following, we report various complexity results concerning forgetting and the logical entailment in this fragment.

Theorem 5 (Entailment on Forgetting) *Let φ and ψ be two CTL_{AF} formulas and V a set of atoms. Then, results:*

- (i) deciding $F_{CTL}(\varphi, V) \models^? \psi$ is co-NP-complete,
- (ii) deciding $\psi \models^? F_{CTL}(\varphi, V)$ is Π_2^P -complete,
- (iii) deciding $F_{CTL}(\varphi, V) \models^? F_{CTL}(\psi, V)$ is Π_2^P -complete.

The following results follow from Theorem 5 and extends them to semantic equivalence.

Corollary 6 *Let φ and ψ be two CTL_{AF} formulas and V a set of atoms. Then*

- (i) deciding $\psi \equiv^? F_{CTL}(\varphi, V)$ is Π_2^P -complete,
- (ii) deciding $F_{CTL}(\varphi, V) \equiv^? \varphi$ is co-NP-complete,
- (iii) deciding $F_{CTL}(\varphi, V) \equiv^? F_{CTL}(\psi, V)$ is Π_2^P -complete.

4 Strongest Necessary and Weakest Sufficient Conditions

In this section, we will give the definition of strongest necessary and weakest sufficient conditions (SNC and WSC, respectively) of a specification in CTL and show that they can be obtained by forgetting under a given initial \mathcal{K} -structure and set V of atoms. We first start with defining these notions of an atomic proposition.

Definition 4 (sufficient and necessary condition) *Let ϕ be a formula (or an initial \mathcal{K} -structure), ψ be a formula, $V \subseteq Var(\phi)$, $q \in Var(\phi) - V$ and $Var(\psi) \subseteq V$.*

- ψ is a necessary condition (NC in short) of q on V under ϕ if $\phi \models q \rightarrow \psi$.
- ψ is a sufficient condition (SC in short) of q on V under ϕ if $\phi \models \psi \rightarrow q$.
- ψ is a strongest necessary condition (SNC in short) of q on V under ϕ if it is a NC of q on V under ϕ and $\phi \models \psi \rightarrow \psi'$ for any NC ψ' of q on V under ϕ .
- ψ is a weakest sufficient condition (WSC in short) of q on V under ϕ if it is a SC of q on V under ϕ and $\phi \models \psi' \rightarrow \psi$ for any SC ψ' of q on V under ϕ .

Note that if both ψ and ψ' are SNC (WSC) of q on V under ϕ , then $Mod(\psi) = Mod(\psi')$, i.e., ψ and ψ' have the same models. In the sense of equivalence the SNC (WSC) is unique (up to equivalence).

Proposition 8 (dual) *Let V, q, φ and ψ are like in Definition 4. The ψ is a SNC (WSC) of q on V under φ iff $\neg\psi$ is a WSC (SNC) of $\neg q$ on V under φ .*

This shows that the SNC and WSC are in fact dual conditions. Under this dual property, we can bother ourselves with only one of them e.g., SNC, and WSC can be obtained easily.

In order to generalise Definition 4 to arbitrary formulas, one can replace q (in the definition) by any formula α , and redefine V as a subset of $Var(\alpha) \cup Var(\phi)$.

It turns out that the previous notion of SNC and WSC for an atomic proposition can be lifted to any formula, or conversely the SNC and WSC of any formula can be reduced to that of a proposition.

Proposition 9 *Let Γ and α be two formulas, $V \subseteq Var(\alpha) \cup Var(\phi)$ and q is a new proposition not in Γ and α . Then, a formula φ of V is the SNC (WSC) of α on V under Γ iff it is the SNC (WSC) of q on V under $\Gamma' = \Gamma \cup \{q \leftrightarrow \alpha\}$.*

The following result establishes the bridge between these two notions which are central to the paper; basically SNC (WSC) and the forgetting in which the former is obtained through the latter.

Algorithm 1: A Model-based Forgetting Procedure

Input: A CTL formula φ and a set V of atoms
Output: $F_{\text{CTL}}(\varphi, V)$

```

1  $T = \emptyset$  // the set of models of  $\varphi$ ;
2  $T' = \emptyset$  // the set of possible initial K-structures;
3  $n = |\mathcal{A}|$ ;
4 for  $i = 1, \dots, 2^n$  do
5   for  $s_j \in \{s_1, \dots, s_i\}$  do
6     Let  $s_j$  be an initial state, construct
        $\mathcal{M} = (S, R, L, s_j)$  by the definition of model
       structure with  $S = \{s_1, \dots, s_i\}$ ;
7     for  $\mathcal{K} \in \mathcal{T}'$  do
8       if  $(\mathcal{M}, s_j) \leftrightarrow_{\text{Var}(\varphi)} \mathcal{K}$  then
9         | Let  $T' \leftarrow T' \cup \{(\mathcal{M}, s_j)\}$ ;
10      end
11    end
12  end
13  for  $(\mathcal{M}, s_0) \in T'$  do
14    if  $(\mathcal{M}, s_0) \models \varphi$  then
15      |  $T \leftarrow T \cup \{(\mathcal{M}, s_0)\}$ ;
16    end
17  end
18 end
19 return  $\bigvee_{(\mathcal{M}', s'_0) \in T} \mathcal{F}_{V'}(\mathcal{M}', s'_0)$ .
```

Theorem 7 Let φ be a formula, $V \subseteq \text{Var}(\varphi)$ and $q \in \text{Var}(\varphi) - V$.

- (i) $F_{\text{CTL}}(\varphi \wedge q, (\text{Var}(\varphi) \cup \{q\}) \setminus V)$ is a SNC of q on V under φ .
- (ii) $\neg F_{\text{CTL}}(\varphi \wedge \neg q, (\text{Var}(\varphi) \cup \{q\}) \setminus V)$ is a WSC of q on V under φ .

As aforementioned, since any initial K-structure can be characterized by a CTL formula, we can obtain the SNC (and its dual WSC) of a target property (a formula) under an initial K-structure by forgetting.

Theorem 8 Let $\mathcal{K} = (\mathcal{M}, s)$ be an initial K-structure with $\mathcal{M} = (S, R, L, s_0)$ on the set \mathcal{A} of atoms, $V \subseteq \mathcal{A}$ and $q \in V' = \mathcal{A} \setminus V$. Then:

- (i) the SNC of q on V under \mathcal{K} is $F_{\text{CTL}}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge q, V')$.
- (ii) the WSC of q on V under \mathcal{K} is $\neg F_{\text{CTL}}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge \neg q, V')$.

Example 3 For the Example 1, the WSC of φ on V under $\mathcal{K} = (\mathcal{M}, s_0)$ is $\neg F_{\text{CTL}}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge (q \leftrightarrow \varphi) \wedge \neg q, \mathcal{A} \setminus V)$.

5 An Algorithm to Compute Forgetting

To compute the forgetting in CTL, we propose a model-based method. Intuitively, the model-based method means that we can compute the forgetting applied to a formula simply by considering all the possible models of that formula.

Next, we give the model-based algorithm i.e., Algorithm 1 to compute the forgetting in CTL. The correctness of this algorithm follows from Lemma 3 and Theorem 2.

Example 4 Let $\varphi = \text{AGAF}(p \wedge r)$, $\mathcal{A} = \{p, r\}$ and $V = \{r\}$. For convenience, we use the label of a state to express the

state and then remove the label function in a model structure. Let $\mathcal{M}_1 = (\{\{p, r\}\}, \{\{p, r\}, \{p, r\}\}, \{p, r\})$ and $\mathcal{M}_2 = (\{\emptyset, \{p, r\}\}, \{\{\emptyset, \{p, r\}\}, \{\{p, r\}, \{p, r\}\}\}, \emptyset)$. The set of models of φ is $\text{Mod}(\varphi) = \{(\mathcal{M}_1, \{p\}), (\mathcal{M}_2, \emptyset), \dots\}$.

Let $\mathcal{M}'_1 = (\{\{p\}\}, \{\{\{p\}, \{p\}\}\}, \{p\})$ and $\mathcal{M}'_2 = (\{\emptyset, \{p\}\}, \{\{\emptyset, \{p\}\}, \{\{p\}, \{p\}\}\}, \emptyset)$. Then we can obtain all the possible initial K-structure that is a model of $F_{\text{CTL}}(\varphi, V)$, i.e., $\text{Mod}(F_{\text{CTL}}(\varphi, V)) = \{\mathcal{K}_1 = (\mathcal{M}'_1, \{p\}), \mathcal{K}_2 = (\mathcal{M}'_2, \emptyset), \dots\}$.

Let $V' = \{p\}$, then $\mathcal{F}_{V'}(\mathcal{K}_1) = p \wedge \text{AG}(p \supset \text{Exp} \wedge \text{Axp})$, and $\mathcal{F}_{V'}(\mathcal{K}_2) = \neg p \wedge \text{AG}(p \supset \text{EX} \neg p \wedge \text{AX} \neg p) \wedge \text{AG}(\neg p \supset \text{Exp} \wedge \text{Axp})$. Similarly, we can obtain the characteristic formula of other models and then the $F_{\text{CTL}}(\varphi, V)$.

Proposition 10 Let φ be a CTL formula and $V \subseteq \mathcal{A}$ with $|\mathcal{A}| = m$ and $|V| = n$. The time and space complexity of Algorithm 1 are $O(2^{m*2^n})$.

6 Conclusion

Summary In this article, we generalized the notion of bisimulation from model structures of Computation Tree Logic (CTL) to model structures with initial states over a given signature V , named V -bisimulation. Based on this new bisimulation, we presented the notion of forgetting for CTL, which enables computing weakest sufficient and strongest necessary conditions of specifications. Further properties and complexity issues in this context were also explored.

Future work Note that, when a transition system \mathcal{M} does not satisfy a specification ϕ , one can evaluate the weakest sufficient condition ψ over a signature V under which \mathcal{M} satisfies ϕ , viz., $\mathcal{M} \models \psi \supset \phi$ and ψ mentions only atoms from V . It is worthwhile to explore how the condition ψ can guide the design of a new transition system \mathcal{M}' satisfying ψ .

Moreover, a further study regarding the the computational complexity for other general fragments is required and part of the future research agenda. Such investigation can be coupled with fine-grained parameterized analysis.

7 Proof

Lemma 5 Let $\mathcal{B}_0, \mathcal{B}_1, \dots$ be the ones in the definition of section 3.1. Then, for each $i \geq 0$,

- (i) $\mathcal{B}_{i+1} \subseteq \mathcal{B}_i$;
- (ii) there is a (smallest) $k \geq 0$ such that $\mathcal{B}_{k+1} = \mathcal{B}_k$;
- (iii) \mathcal{B}_i is reflexive, symmetric and transitive.

Proof: (i) Base: it is clear for $i = 0$ by the above definition.

Step: suppose it holds for $i = n$, i.e., $\mathcal{B}_{n+1} \subseteq \mathcal{B}_n$.

$(s, s') \in \mathcal{B}_{n+2}$

\Rightarrow (a) $(s, s') \in \mathcal{B}_0$, (b) for every $(s, s_1) \in R$, there is $(s', s'_1) \in R'$ such that $(s_1, s'_1) \in \mathcal{B}_{n+1}$, and (c) for every $(s', s'_1) \in R'$, there is $(s, s_1) \in R$ such that $(s_1, s'_1) \in \mathcal{B}_{n+1}$

\Rightarrow (a) $(s, s') \in \mathcal{B}_0$, (b) for every $(s, s_1) \in R$, there is $(s', s'_1) \in R'$ such that $(s_1, s'_1) \in \mathcal{B}_n$ by inductive assumption, and (c) for every $(s', s'_1) \in R'$, there is $(s, s_1) \in R$ such that $(s_1, s'_1) \in \mathcal{B}_n$ by inductive assumption

$\Rightarrow (s, s') \in \mathcal{B}_{n+1}$.

(ii) and (iii) are evident from (i) and the definition of \mathcal{B}_i . ■

Lemma 1 The relation \leftrightarrow_V is an equivalence relation.

Proof: It is clear from Lemma 5 (ii) such that there is a $k \geq 0$ where $\mathcal{B}_k = \mathcal{B}_{k+1}$ which is \leftrightarrow_V , and it is reflexive, symmetric and transitive by (iii). ■

Proposition 1 Let $i \in \{1, 2\}$, $V_1, V_2 \subseteq \mathcal{A}$, s'_i s be two states and π'_i s be two paths, and $\mathcal{K}_i = (\mathcal{M}_i, s_i)$ ($i = 1, 2, 3$) be K-structures such that $\mathcal{K}_1 \leftrightarrow_{V_1} \mathcal{K}_2$ and $\mathcal{K}_2 \leftrightarrow_{V_2} \mathcal{K}_3$. Then:

- (i) $s'_1 \leftrightarrow_{V_1} s'_2$ ($i = 1, 2$) implies $s'_1 \leftrightarrow_{V_1 \cup V_2} s'_2$;
- (ii) $\pi'_1 \leftrightarrow_{V_1} \pi'_2$ ($i = 1, 2$) implies $\pi'_1 \leftrightarrow_{V_1 \cup V_2} \pi'_2$;
- (iii) for each path π_{s_1} of \mathcal{M}_1 there is a path π_{s_2} of \mathcal{M}_2 such that $\pi_{s_1} \leftrightarrow_{V_1} \pi_{s_2}$, and vice versa;
- (iv) $\mathcal{K}_1 \leftrightarrow_{V_1 \cup V_2} \mathcal{K}_3$;
- (v) If $V_1 \subseteq V_2$ then $\mathcal{K}_1 \leftrightarrow_{V_2} \mathcal{K}_2$.

Proof: In order to distinguish the relations $\mathcal{B}_0, \mathcal{B}_1, \dots$ for different set $V \subseteq \mathcal{A}$, by \mathcal{B}_i^V we mean the relation $\mathcal{B}_1, \mathcal{B}_2, \dots$ for $V \subseteq \mathcal{A}$. Denote as $\mathcal{B}_0, \mathcal{B}_1, \dots$ when the underlying set V is clear from the context. Moreover, for the ease of notation, we will refer to \leftrightarrow_V by \mathcal{B} (i.e., without subindex).

(i) Base: it is clear for $n = 0$.

Step: For $n > 0$, supposing if $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i^{V_1}$ and $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i^{V_2}$ then $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i^{V_1 \cup V_2}$ for all $0 \leq i \leq n$. We will show that if $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}^{V_1}$ and $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}^{V_2}$ then $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}^{V_1 \cup V_2}$.

(a) It is evident that $L_1(s_1) - (V_1 \cup V_2) = L_2(s_2) - (V_1 \cup V_2)$.
(b) We will show that for each $(s_1, s_1^1) \in R_1$ there is a $(s_2, s_2^1) \in R_2$ such that $(s_1^1, s_2^1) \in \mathcal{B}_{n-1}^{V_1 \cup V_2}$. There is $(\mathcal{K}_1^1, \mathcal{K}_2^1) \in \mathcal{B}_{n-1}^{V_1 \cup V_2}$ due to $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_n^{V_1 \cup V_2}$ by inductive assumption. Then we only need to prove for each $(s_1^1, s_1^2) \in R_1$ there is a $(s_2^1, s_2^2) \in R_2$ such that $(\mathcal{K}_1^1, \mathcal{K}_2^2) \in \mathcal{B}_{n-2}^{V_1 \cup V_2}$ and for each $(s_2^1, s_2^2) \in R_2$ there is a $(s_1^1, s_1^2) \in R_1$ such that $(\mathcal{K}_1^2, \mathcal{K}_2^2) \in \mathcal{B}_{n-2}^{V_1 \cup V_2}$. Therefore, we only need to prove that for each $(s_1^n, s_1^{n+1}) \in R_1$ there is a $(s_2^n, s_2^{n+1}) \in R_2$ such that $(\mathcal{K}_1^{n+1}, \mathcal{K}_2^{n+1}) \in \mathcal{B}_0^{V_1 \cup V_2}$ and for each $(s_2^n, s_2^{n+1}) \in R_2$ there is a $(s_1^n, s_1^{n+1}) \in R_1$ such that $(\mathcal{K}_1^{n+1}, \mathcal{K}_2^{n+1}) \in \mathcal{B}_0^{V_1 \cup V_2}$. It is apparent that $L_1(s_1^{n+1}) - (V_1 \cup V_2) = L_1(s_2^{n+1}) - (V_1 \cup V_2)$ due to $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}^{V_1}$ and $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{n+1}^{V_2}$. Where $\mathcal{K}_i^j = (\mathcal{M}_i, s_i^j)$ with $i \in \{1, 2\}$ and $0 < j \leq n+1$.

(c) It is similar with (b).

(ii) It is clear from (i).

(iii) The following property show our result directly. Let $V \subseteq \mathcal{A}$ and $\mathcal{K}_i = (\mathcal{M}_i, s_i)$ ($i = 1, 2$) be K-structures. Then $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}$ if and only if

- (a) $L_1(s_1) - V = L_2(s_2) - V$,
- (b) for every $(s_1, s'_1) \in R_1$, there is $(s_2, s'_2) \in R_2$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}$, and
- (c) for every $(s_2, s'_2) \in R_2$, there is $(s_1, s'_1) \in R_1$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}$,

where $\mathcal{K}'_i = (\mathcal{M}_i, s'_i)$ with $i \in \{1, 2\}$.

We prove it from the following two aspects:

(\Rightarrow) (a) It is apparent that $L_1(s_1) - V = L_2(s_2) - V$; (b) $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}$ iff $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i$ for all $i \geq 0$, then for each $(s_1, s'_1) \in R_1$, there is a $(s_2, s'_2) \in R_2$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in$

\mathcal{B}_{i-1} for all $i > 0$ and then $L_1(s'_1) - V = L_2(s'_2) - V$. Therefore, $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}$. (c) This is similar with (b).

(\Leftarrow) (a) $L_1(s_1) - V = L_2(s_2) - V$ implies that $(s_1, s_2) \in \mathcal{B}_0$; (b) Condition (ii) implies that for every $(s_1, s'_1) \in R_1$, there is $(s_2, s'_2) \in R_2$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}_i$ for all $i \geq 0$; (c) Condition (iii) implies that for every $(s_2, s'_2) \in R_2$, there is $(s_1, s'_1) \in R_1$ such that $(\mathcal{K}'_1, \mathcal{K}'_2) \in \mathcal{B}_i$ for all $i \geq 0 \Rightarrow (\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i$ for all $i \geq 0 \Rightarrow (\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}$.

(iv) Let $\mathcal{M}_i = (S_i, R_i, L_i, s_i)$ ($i = 1, 2, 3$), $s_1 \leftrightarrow_{V_1} s_2$ via a binary relation \mathcal{B} , and $s_2 \leftrightarrow_{V_2} s_3$ via a binary relation \mathcal{B}'' . Let $\mathcal{B}' = \{(w_1, w_3) | (w_1, w_2) \in \mathcal{B} \text{ and } (w_2, w_3) \in \mathcal{B}_2\}$. It's apparent that $(s_1, s_3) \in \mathcal{B}'$. We prove \mathcal{B}' is a $V_1 \cup V_2$ -bisimulation containing (s_1, s_3) from the (a), (b) and (c) of the previous step (iii) of X -bisimulation (where X is a set of atoms). For all $(w_1, w_3) \in \mathcal{B}'$:

- (a) there is $w_2 \in S_2$ such that $(w_1, w_2) \in \mathcal{B}$ and $(w_2, w_3) \in \mathcal{B}''$, and $\forall q \notin V_1$, $q \in L_1(w_1)$ iff $q \in L_2(w_2)$ by $w_1 \leftrightarrow_{V_1} w_2$ and $\forall q' \notin V_2$, $q' \in L_2(w_2)$ iff $q' \in L_3(w_3)$ by $w_2 \leftrightarrow_{V_2} w_3$. Then we have $\forall r \notin V_1 \cup V_2$, $r \in L_1(w_1)$ iff $r \in L_3(w_3)$.
- (b) if $(w_1, u_1) \in \mathcal{R}_1$, then $\exists u_2 \in S_2$ such that $(w_2, u_2) \in \mathcal{R}_2$ and $(u_1, u_2) \in \mathcal{B}$ (due to $(w_1, w_2) \in \mathcal{B}$ and $(w_2, w_3) \in \mathcal{B}''$ by the definition of \mathcal{B}'); and then $\exists u_3 \in S_3$ such that $(w_3, u_3) \in \mathcal{R}_3$ and $(u_2, u_3) \in \mathcal{B}''$, hence $(u_1, u_3) \in \mathcal{B}'$ by the definition of \mathcal{B}' .
- (c) if $(w_3, u_3) \in \mathcal{R}_3$, then $\exists u_2 \in S_2$ such that $(w_2, u_2) \in \mathcal{R}_2$ and $(u_2, u_3) \in \mathcal{B}_2$; and then $\exists u_1 \in S_1$ such that $(w_1, u_1) \in \mathcal{R}_1$ and $(u_1, u_2) \in \mathcal{B}$, hence $(u_1, u_3) \in \mathcal{B}'$ by the definition of \mathcal{B}' .

(v) Let $\mathcal{K}_{i,j} = (\mathcal{M}_i, s_{i,j})$ and $(s_{i,k}, s_{i,k+1}) \in R_i$ mean that $s_{i,k+1}$ is the $(k+2)$ -th node in the path $(s_i, s_{i,1}, s_{i,2}, \dots, s_{i,k+1}, \dots)$ ($i = 1, 2$). We will show that $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_n^{V_2}$ for all $n \geq 0$ inductively.

Base: $L_1(s_1) - V_1 = L_2(s_2) - V_1 \Rightarrow \forall q \in \mathcal{A} - V_1$ there is $q \in L_1(s_1)$ iff $q \in L_2(s_2) \Rightarrow \forall q \in \mathcal{A} - V_2$ there is $q \in L_1(s_1)$ iff $q \in L_2(s_2)$ due to $V_1 \subseteq V_2 \Rightarrow L_1(s_1) - V_2 = L_2(s_2) - V_2$, i.e., $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_0^{V_2}$.

Step: Supposing that $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_i^{V_2}$ for all $0 \leq i \leq k$ ($k > 0$), we will show $(\mathcal{K}_1, \mathcal{K}_2) \in \mathcal{B}_{k+1}^{V_2}$.

- (a) It is apparent that $L_1(s_1) - V_2 = L_2(s_2) - V_2$ by base.
- (b) $\forall (s_1, s_{1,1}) \in R_1$, we will show that there is a $(s_2, s_{2,1}) \in R_2$ $(\mathcal{K}_{1,1}, \mathcal{K}_{2,1}) \in \mathcal{B}_k^{V_2}$. $(\mathcal{K}_{1,1}, \mathcal{K}_{2,1}) \in \mathcal{B}_{k-1}^{V_2}$ by inductive assumption, we need only to prove the following points:
 - (a) $\forall (s_{1,k}, s_{1,k+1}) \in R_1$ there is a $(s_{2,k}, s_{2,k+1}) \in R_2$ $(\mathcal{K}_{1,k+1}, \mathcal{K}_{2,k+1}) \in \mathcal{B}_0^{V_2}$ due to $(\mathcal{K}_{1,1}, \mathcal{K}_{2,1}) \in \mathcal{B}_k^{V_1}$. It is easy to see that $L_1(s_{1,k+1}) - V_1 = L_1(s_{2,k+1}) - V_1$, then there is $L_1(s_{1,k+1}) - V_2 = L_1(s_{2,k+1}) - V_2$. Therefore, $(\mathcal{K}_{1,k+1}, \mathcal{K}_{2,k+1}) \in \mathcal{B}_0^{V_2}$.
 - (b) $\forall (s_{2,k}, s_{2,k+1}) \in R_1$ there is a $(s_{1,k}, s_{1,k+1}) \in R_1$ $(\mathcal{K}_{1,k+1}, \mathcal{K}_{2,k+1}) \in \mathcal{B}_0^{V_2}$ due to $(\mathcal{K}_{1,1}, \mathcal{K}_{2,1}) \in \mathcal{B}_k^{V_1}$. This can be proved as (a).
- (c) $\forall (s_2, s_{2,1}) \in R_1$, we will show that there is a $(s_1, s_{1,1}) \in R_2$ $(\mathcal{K}_{1,1}, \mathcal{K}_{2,1}) \in \mathcal{B}_k^{V_2}$. This can be proved as (ii).

■

Theorem 1 Let $V \subseteq \mathcal{A}$, \mathcal{K}_i ($i = 1, 2$) be two K-structures such that $\mathcal{K}_1 \leftrightarrow_V \mathcal{K}_2$ and ϕ a formula with $\text{IR}(\phi, V)$. Then $\mathcal{K}_1 \models \phi$ if and only if $\mathcal{K}_2 \models \phi$.

Proof: This theorem can be proved by inducting on the formula ϕ and supposing $\text{Var}(\phi) \cap V = \emptyset$.

Here we only prove the only-if direction. The other direction can be similarly proved.

Case $\phi = p$ where $p \in \mathcal{A} - V$:

$(\mathcal{M}, s) \models \phi$ iff $p \in L(s)$ (by the definition of satisfiability)
 $\Leftrightarrow p \in L'(s')$ ($s \leftrightarrow_V s'$)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$

Case $\phi = \neg\psi$:

$(\mathcal{M}, s) \models \phi$ iff $(\mathcal{M}, s) \not\models \psi$
 $\Leftrightarrow (\mathcal{M}', s') \not\models \psi$ (induction hypothesis)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$

Case $\phi = \psi_1 \vee \psi_2$:

$(\mathcal{M}, s) \models \phi$
 $\Leftrightarrow (\mathcal{M}, s) \models \psi_1$ or $(\mathcal{M}, s) \models \psi_2$
 $\Leftrightarrow (\mathcal{M}', s') \models \psi_1$ or $(\mathcal{M}', s') \models \psi_2$ (induction hypothesis)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$

Case $\phi = \text{EX}\psi$:

$\mathcal{M}, s \models \phi$
 \Leftrightarrow There is a path $\pi = (s, s_1, \dots)$ such that $\mathcal{M}, s_1 \models \psi$
 \Leftrightarrow There is a path $\pi' = (s', s'_1, \dots)$ such that $\pi \leftrightarrow_V \pi'$ ($s \leftrightarrow_V s'$, Proposition 1)
 $\Leftrightarrow s_1 \leftrightarrow_V s'_1$ ($\pi \leftrightarrow_V \pi'$)
 $\Leftrightarrow (\mathcal{M}', s'_1) \models \psi$ (induction hypothesis)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$

Case $\phi = \text{EG}\psi$:

$\mathcal{M}, s \models \phi$
 \Leftrightarrow There is a path $\pi = (s = s_0, s_1, \dots)$ such that for each $i \geq 0$ there is $(\mathcal{M}, s_i) \models \psi$
 \Leftrightarrow There is a path $\pi' = (s' = s'_0, s'_1, \dots)$ such that $\pi \leftrightarrow_V \pi'$ ($s \leftrightarrow_V s'$, Proposition 1)
 $\Leftrightarrow s_i \leftrightarrow_V s'_i$ for each $i \geq 0$ ($\pi \leftrightarrow_V \pi'$)
 $\Leftrightarrow (\mathcal{M}', s'_i) \models \psi$ for each $i \geq 0$ (induction hypothesis)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$

Case $\phi = \text{E}[\psi_1 \cup \psi_2]$:

$\mathcal{M}, s \models \phi$
 \Leftrightarrow There is a path $\pi = (s = s_0, s_1, \dots)$ such that there is $i \geq 0$ such that $(\mathcal{M}, s_i) \models \psi_2$, and for all $0 \leq j < i$, $(\mathcal{M}, s_j) \models \psi_1$
 \Leftrightarrow There is a path $\pi' = (s = s'_0, s'_1, \dots)$ such that $\pi \leftrightarrow_V \pi'$ ($s \leftrightarrow_V s'$, Proposition 1)
 $\Leftrightarrow (\mathcal{M}', s'_i) \models \psi_2$, and for all $0 \leq j < i$ $(\mathcal{M}', s'_j) \models \psi_1$ (induction hypothesis)
 $\Leftrightarrow (\mathcal{M}', s') \models \phi$ ■

Proposition 2 Let $V \subseteq \mathcal{A}$ and (\mathcal{M}_i, s_i) ($i = 1, 2$) be two K-structures. Then

$(s_1, s_2) \in \mathcal{B}_n$ iff $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for every $0 \leq j \leq n$.

Proof: We will prove this from two aspects:

(\Rightarrow) If $(s_1, s_2) \in \mathcal{B}_n$, then $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for all $0 \leq j \leq n$. $(s, s') \in \mathcal{B}_n$ implies both roots of $\text{Tr}_n(s_1)$ and $\text{Tr}_n(s_2)$ have the same atoms except those atoms in V . Besides, for any $s_{1,1}$ with $(s_1, s_{1,1}) \in R_1$, there is a

$s_{2,1}$ with $(s_2, s_{2,1}) \in R_2$ s.t. $(s_{1,1}, s_{2,1}) \in \mathcal{B}_{n-1}$ and vice versa. Then we have $\text{Tr}_1(s_1) \leftrightarrow_V \text{Tr}_1(s_2)$. Therefore, $\text{Tr}_n(s_1) \leftrightarrow_V \text{Tr}_n(s_2)$ by use such method recursively, and then $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for all $0 \leq j \leq n$.

(\Leftarrow) If $\text{Tr}_j(s_1) \leftrightarrow_V \text{Tr}_j(s_2)$ for all $0 \leq j \leq n$, then $(s_1, s_2) \in \mathcal{B}_n$. $\text{Tr}_0(s_1) \leftrightarrow_V \text{Tr}_0(s_2)$ implies $L(s_1) - V = L(s_2) - V$ and then $(s, s') \in \mathcal{B}_0$. $\text{Tr}_1(s_1) \leftrightarrow_V \text{Tr}_1(s_2)$ implies $L(s_1) - V = L(s_2) - V$ and for every successors s of the root of one, it is possible to find a successor of the root of the other s' such that $(s, s') \in \mathcal{B}_0$. Therefore $(s_1, s_2) \in \mathcal{B}_1$, and then we will have $(s_1, s_2) \in \mathcal{B}_n$ by use such method recursively. ■

Proposition 3 Let $V \subseteq \mathcal{A}$, \mathcal{M} be a model structure and $s, s' \in S$ such that $s \not\leftrightarrow_V s'$. There exists a least k such that $\text{Tr}_k(s)$ and $\text{Tr}_k(s')$ are not V -bisimilar.

Proof: If $s \not\leftrightarrow_V s'$, then there exists a least constant k such that $(s_i, s_j) \notin \mathcal{B}_k$, and then there is a least constant m ($m \leq k$) such that $\text{Tr}_m(s_i)$ and $\text{Tr}_m(s_j)$ are not V -corresponding by Proposition 2. Let $c = m$, the lemma is proved. ■

Lemma 2 Let $V \subseteq \mathcal{A}$, \mathcal{M} and \mathcal{M}' be two model structures, $s \in S$, $s' \in S'$ and $n \geq 0$. If $\text{Tr}_n(s) \leftrightarrow_{\overline{V}} \text{Tr}_n(s')$, then $\mathcal{F}_V(\text{Tr}_n(s)) \equiv \mathcal{F}_V(\text{Tr}_n(s'))$.

Proof: This result can be proved by inducting on n .

Base. It is apparent that for any $s_n \in S$ and $s'_n \in S'$, if $\text{Tr}_0(s_n) \leftrightarrow_{\overline{V}} \text{Tr}_0(s'_n)$ then $\mathcal{F}_V(\text{Tr}_0(s_n)) \equiv \mathcal{F}_V(\text{Tr}_0(s'_n))$ due to $L(s_n) - \overline{V} = L(s'_n) - \overline{V}$ by known.

Step. Supposing that for $k = m$ ($0 < m \leq n$) there is if $\text{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k}(s'_k)$ then $\mathcal{F}_V(\text{Tr}_{n-k}(s_k)) \equiv \mathcal{F}_V(\text{Tr}_{n-k}(s'_k))$, then we will show if $\text{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k+1}(s'_{k-1})$ then $\mathcal{F}_V(\text{Tr}_{n-k+1}(s_{k-1})) \equiv \mathcal{F}_V(\text{Tr}_{n-k+1}(s'_{k-1}))$. Apparent that:

$$\mathcal{F}_V(\text{Tr}_{n-k+1}(s_{k-1})) = \left(\bigwedge_{(s_{k-1}, s_k) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_{n-k}(s_k)) \right) \wedge \text{AX} \left(\bigvee_{(s_{k-1}, s_k) \in R} \mathcal{F}_V(\text{Tr}_{n-k}(s_k)) \right) \wedge \mathcal{F}_V(\text{Tr}_0(s_{k-1}))$$

$$\mathcal{F}_V(\text{Tr}_{n-k+1}(s'_{k-1})) = \left(\bigwedge_{(s'_{k-1}, s'_k) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_{n-k}(s'_k)) \right) \wedge \text{AX} \left(\bigvee_{(s'_{k-1}, s'_k) \in R} \mathcal{F}_V(\text{Tr}_{n-k}(s'_k)) \right) \wedge \mathcal{F}_V(\text{Tr}_0(s'_{k-1}))$$

by the definition of characterizing formula of the computation tree. Then we have for any $(s_{k-1}, s_k) \in R$ there is $(s'_{k-1}, s'_k) \in R'$ such that $\text{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k}(s'_k)$ by $\text{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k+1}(s'_{k-1})$. Besides, for any $(s'_{k-1}, s'_k) \in R'$ there is $(s_{k-1}, s_k) \in R$ such that $\text{Tr}_{n-k}(s_k) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k}(s'_k)$ by $\text{Tr}_{n-k+1}(s_{k-1}) \leftrightarrow_{\overline{V}} \text{Tr}_{n-k+1}(s'_{k-1})$. Therefore, we have $\mathcal{F}_V(\text{Tr}_{n-k+1}(s_{k-1})) \equiv \mathcal{F}_V(\text{Tr}_{n-k+1}(s'_{k-1}))$ by induction hypothesis. ■

Theorem 2 Given $V \subseteq \mathcal{A}$, let $\mathcal{M} = (S, R, L, s_0)$ and $\mathcal{M}' = (S', R', L', s'_0)$ be two model structures. Then,

(i) $(\mathcal{M}', s'_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$ iff $(\mathcal{M}, s_0) \leftrightarrow_{\overline{V}} (\mathcal{M}', s'_0)$;

(ii) $s_0 \leftrightarrow_{\overline{V}} s'_0$ implies $\mathcal{F}_V(\mathcal{M}, s_0) \equiv \mathcal{F}_V(\mathcal{M}', s'_0)$.

In order to prove Theorem 2, we prove the following two lemmas at first.

Lemma 6 Let $V \subseteq \mathcal{A}$, $\mathcal{M} = (S, R, L, s_0)$ and $\mathcal{M}' = (S', R', L', s'_0)$ be two model structures, $s \in S$, $s' \in S'$ and $n \geq 0$.

(i) $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_n(s))$.

(ii) If $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_n(s'))$ then $\text{Tr}_n(s) \leftrightarrow_{\overline{V}} \text{Tr}_n(s')$.

Proof: (i) It is apparent from the definition of $\mathcal{F}_V(\text{Tr}_n(s))$. Base. It is apparent that $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_0(s))$.

Step. For $k \geq 0$, supposing the result talked in (i) is correct in $k-1$, we will show that $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_{k+1}(s))$, i.e.:

$$(\mathcal{M}, s) \models \left(\bigwedge_{(s, s') \in R} \text{EXT}(s') \right) \wedge \text{AX} \left(\bigvee_{(s, s') \in R} T(s') \right) \wedge \mathcal{F}_V(\text{Tr}_0(s)).$$

Where $T(s') = \mathcal{F}_V(\text{Tr}_k(s'))$. It is apparent that $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_0(s))$ by Base. It is apparent that for any $(s, s') \in R$, there is $(\mathcal{M}, s') \models \mathcal{F}_V(\text{Tr}_k(s'))$ by inductive assumption. Then we have $(\mathcal{M}, s) \models \text{EX}\mathcal{F}_V(\text{Tr}_k(s'))$, and then $(\mathcal{M}, s) \models \left(\bigwedge_{(s, s') \in R} \text{EX}\mathcal{F}_V(\text{Tr}_k(s')) \right)$. Similarly, we have that for any $(s, s') \in R$, there is $(\mathcal{M}, s') \models \bigvee_{(s, s'') \in R} \mathcal{F}_V(\text{Tr}_k(s''))$.

Therefore, $(\mathcal{M}, s) \models \text{AX} \left(\bigvee_{(s, s'') \in R} \mathcal{F}_V(\text{Tr}_k(s'')) \right)$.

(ii) **Base.** If $n = 0$, then $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_0(s'))$ implies $L(s) - \overline{V} = L'(s') - \overline{V}$. Hence, $\text{Tr}_0(s) \leftrightarrow_{\overline{V}} \text{Tr}_0(s')$.

Step. Supposing $n > 0$ and the result talked in (ii) is correct in $n-1$.

(a) It is easy to see that $L(s) - \overline{V} = L'(s') - \overline{V}$.

(b) We will show that for each $(s, s_1) \in R$, there is a $(s', s'_1) \in R'$ such that $\text{Tr}_{n-1}(s_1) \leftrightarrow_{\overline{V}} \text{Tr}_{n-1}(s'_1)$. Since $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_n(s'))$, then $(\mathcal{M}, s) \models \text{AX} \left(\bigvee_{(s', s'_1) \in R} \mathcal{F}_V(\text{Tr}_{n-1}(s'_1)) \right)$. Therefore, for each $(s, s_1) \in R$ there is a $(s', s'_1) \in R'$ such that $(\mathcal{M}, s_1) \models \mathcal{F}_V(\text{Tr}_{n-1}(s'_1))$. Hence, $\text{Tr}_{n-1}(s_1) \leftrightarrow_{\overline{V}} \text{Tr}_{n-1}(s'_1)$ by inductive hypothesis.

(c) We will show that for each $(s', s'_1) \in R'$ there is a $(s, s_1) \in R$ such that $\text{Tr}_{n-1}(s'_1) \leftrightarrow_{\overline{V}} \text{Tr}_{n-1}(s_1)$. Since $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_n(s'))$, then $(\mathcal{M}, s) \models \bigwedge_{(s', s'_1) \in R'} \text{EX}\mathcal{F}_V(\text{Tr}_{n-1}(s'_1))$. Therefore, for each $(s', s'_1) \in R'$ there is a $(s, s_1) \in R$ such that $(\mathcal{M}, s_1) \models \mathcal{F}_V(\text{Tr}_{n-1}(s'_1))$. Hence, $\text{Tr}_{n-1}(s_1) \leftrightarrow_{\overline{V}} \text{Tr}_{n-1}(s'_1)$ by inductive hypothesis. ■

A consequence of the previous lemma is:

Lemma 7 Let $V \subseteq \mathcal{A}$, $\mathcal{M} = (S, R, L, s_0)$ a model structure, $k = \text{ch}(\mathcal{M}, V)$ and $s \in S$.

- $(\mathcal{M}, s) \models \mathcal{F}_V(\text{Tr}_k(s))$, and
- for each $s' \in S$, $(\mathcal{M}, s) \leftrightarrow_{\overline{V}} (\mathcal{M}, s')$ if and only if $(\mathcal{M}, s') \models \mathcal{F}_V(\text{Tr}_k(s))$.

Proof: Let $\phi = \mathcal{F}_V(\text{Tr}_k(s))$, where k is the V-characteristic number of \mathcal{M} . $(\mathcal{M}, s) \models \phi$ by the definition of \mathcal{F} , and then $\forall s' \in S$, if $s \leftrightarrow_{\overline{V}} s'$ there is $(\mathcal{M}, s') \models \phi$ by Theorem 1 due to $\text{IR}(\phi, \mathcal{A} - V)$. Supposing $(\mathcal{M}, s') \models \phi$, if $s \not\leftrightarrow_{\overline{V}} s'$, then $\text{Tr}_k(s) \not\leftrightarrow_{\overline{V}} \text{Tr}_k(s')$, and then $(\mathcal{M}, s') \not\models \phi$ by Lemma 6, a contradiction. ■

Now we are in the position of proving Theorem 2.

Proof: (i) Let $\mathcal{F}_V(\mathcal{M}, s_0)$ be the characterizing formula of (\mathcal{M}, s_0) on V . It is apparent that $\text{IR}(\mathcal{F}_V(\mathcal{M}, s_0), \overline{V})$. We will show that $(\mathcal{M}, s_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$ at first.

It is apparent that $(\mathcal{M}, s_0) \models \mathcal{F}_V(\text{Tr}_c(s_0))$ by Lemma 6. We must show that $(\mathcal{M}, s_0) \models \bigwedge_{s \in S} G(\mathcal{M}, s)$. Let

$\mathcal{X} = \mathcal{F}_V(\text{Tr}_c(s)) \rightarrow \left(\bigwedge_{(s, s_1) \in R} \text{EX}\mathcal{F}_V(\text{Tr}_c(s_1)) \right) \wedge \text{AX} \left(\bigvee_{(s, s_1) \in R} \mathcal{F}_V(\text{Tr}_c(s_1)) \right)$, we will show $\forall s \in S$, $(\mathcal{M}, s_0) \models G(\mathcal{M}, s)$. Where $G(\mathcal{M}, s) = \text{AG}\mathcal{X}$. There are two cases we should consider:

- If $(\mathcal{M}, s_0) \not\models \mathcal{F}_V(\text{Tr}_c(s))$, it is apparent that $(\mathcal{M}, s_0) \models \mathcal{X}$;
- If $(\mathcal{M}, s_0) \models \mathcal{F}_V(\text{Tr}_c(s))$:
 $(\mathcal{M}, s_0) \models \mathcal{F}_V(\text{Tr}_c(s))$
 $\Rightarrow s_0 \leftrightarrow_{\overline{V}} s$ by the definition of characteristic number and Lemma 7.

For each $(s, s_1) \in R$ there is:

$$\begin{aligned} (\mathcal{M}, s_1) &\models \mathcal{F}_V(\text{Tr}_c(s_1)) & (s_1 \leftrightarrow_{\overline{V}} s_1) \\ \Rightarrow (\mathcal{M}, s) &\models \bigwedge_{(s, s_1) \in R} \text{EX}\mathcal{F}_V(\text{Tr}_c(s_1)) \\ \Rightarrow (\mathcal{M}, s_0) &\models \bigwedge_{(s, s_1) \in R} \text{EX}\mathcal{F}_V(\text{Tr}_c(s_1)) & \text{(by)} \\ \text{IR}(\bigwedge_{(s, s_1) \in R} \text{EX}\mathcal{F}_V(\text{Tr}_c(s_1)), \overline{V}), &s_0 \leftrightarrow_{\overline{V}} s. \end{aligned}$$

For each (s, s_1) there is:

$$\begin{aligned} \mathcal{M}, s_1 &\models \bigvee_{(s, s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \\ \Rightarrow (\mathcal{M}, s) &\models \text{AX} \left(\bigvee_{(s, s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) \\ \Rightarrow (\mathcal{M}, s_0) &\models \text{AX} \left(\bigvee_{(s, s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) & \text{(by)} \\ \text{IR}(\text{AX} \left(\bigvee_{(s, s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \right), \overline{V}), &s_0 \leftrightarrow_{\overline{V}} s \\ \Rightarrow (\mathcal{M}, s_0) &\models \mathcal{X}. \end{aligned}$$

For any other states s' which can reach from s_0 can be proved similarly, i.e., $(\mathcal{M}, s') \models \mathcal{X}$. Therefore, $\forall s \in S$, $(\mathcal{M}, s_0) \models G(\mathcal{M}, s)$, and then $(\mathcal{M}, s_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$.

We will prove this theorem from the following two aspects:

(\Leftarrow) If $s_0 \leftrightarrow_{\overline{V}} s'_0$, then $(\mathcal{M}', s'_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$. Since $(\mathcal{M}, s_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$ and $\text{IR}(\mathcal{F}_V(\mathcal{M}, s_0), \overline{V})$, hence $(\mathcal{M}', s'_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$ by Theorem 1.

(\Rightarrow) If $(\mathcal{M}', s'_0) \models \mathcal{F}_V(\mathcal{M}, s_0)$, then $s_0 \leftrightarrow_{\overline{V}} s'_0$. We will prove this by showing that $\forall n \geq 0$, $\text{Tr}_n(s_0) \leftrightarrow_{\overline{V}} \text{Tr}_n(s'_0)$.

Base. It is apparent that $\text{Tr}_0(s_0) \equiv \text{Tr}_0(s'_0)$.

Step. Supposing $\text{Tr}_k(s_0) \leftrightarrow_{\overline{V}} \text{Tr}_k(s'_0)$ ($k > 0$), we will prove $\text{Tr}_{k+1}(s_0) \leftrightarrow_{\overline{V}} \text{Tr}_{k+1}(s'_0)$. We should only show that $\text{Tr}_1(s_k) \leftrightarrow_{\overline{V}} \text{Tr}_1(s'_k)$. Where $(s_0, s_1), (s_1, s_2), \dots, (s_{k-1}, s_k) \in R$ and $(s'_0, s'_1), (s'_1, s'_2), \dots, (s'_{k-1}, s'_k) \in R'$, i.e., $s_{i+1} (s'_{i+1})$ is an immediate successor of $s_i (s'_i)$ for all $0 \leq i \leq k-1$.

(a) It is apparent that $L(s_k) - \overline{V} = L'(s'_k) - \overline{V}$ by inductive assumption.

Before talking about the other points, note the following fact that:

$$\begin{aligned} (\mathcal{M}', s'_0) &\models \mathcal{F}_V(\mathcal{M}, s_0) \\ \Rightarrow \forall s' \in S', (\mathcal{M}', s') &\models \mathcal{F}_V(\text{Tr}_c(s)) \rightarrow \\ &\left(\bigwedge_{(s, s_1) \in R} \text{EX}\mathcal{F}_V(\text{Tr}_c(s_1)) \right) \end{aligned} \quad \wedge$$

$$\begin{aligned}
& \text{AX} \left(\bigvee_{(s,s_1) \in R} \mathcal{F}_V(\text{Tr}_c(s_1)) \right) \text{ for any } s \in S. & \text{(fact)} \\
\text{(I)} \quad (\mathcal{M}', s'_0) & \models \mathcal{F}_V(\text{Tr}_c(s_0)) & \rightarrow \\
& \left(\bigwedge_{(s_0,s_1) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_1)) \right) & \wedge \\
& \text{AX} \left(\bigvee_{(s_0,s_1) \in R} \mathcal{F}_V(\text{Tr}_c(s_1)) \right) & \text{(fact)} \\
\text{(II)} \quad (\mathcal{M}', s'_0) & \models \mathcal{F}_V(\text{Tr}_c(s_0)) & \text{(known)} \\
\text{(III)} \quad (\mathcal{M}', s'_0) & \models \left(\bigwedge_{(s_0,s_1) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_1)) \right) & \wedge \\
& \text{AX} \left(\bigvee_{(s_0,s_1) \in R} \mathcal{F}_V(\text{Tr}_c(s_1)) \right) & \text{((I),(II))}
\end{aligned}$$

(b) We will show that for each $(s_k, s_{k+1}) \in R$ there is a $(s'_k, s'_{k+1}) \in R'$ such that $L(s_{k+1}) - \bar{V} = L'(s'_{k+1}) - \bar{V}$.

$$\begin{aligned}
\text{(1)} \quad (\mathcal{M}', s'_0) & \models \bigwedge_{(s_0,s_1) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_1)) & \text{(III)} \\
\text{(2)} \quad \forall (s_0, s_1) \in R, \exists (s'_0, s'_1) \in R' & (\mathcal{M}', s'_1) \models \mathcal{F}_V(\text{Tr}_c(s_1)) \\
\text{(3)} \quad \text{Tr}_c(s_1) \leftrightarrow_{\bar{V}} \text{Tr}_c(s'_1) & \text{((2), Lemma 6)} \\
\text{(4)} \quad L(s_1) - \bar{V} = L'(s'_1) - \bar{V} & \text{((3), } c \geq 0) \\
\text{(5)} \quad (\mathcal{M}', s'_1) & \models \mathcal{F}_V(\text{Tr}_c(s_1)) & \rightarrow \\
& \left(\bigwedge_{(s_1,s_2) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) & \wedge \\
& \text{AX} \left(\bigvee_{(s_1,s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) & \text{(fact)} \\
\text{(6)} \quad (\mathcal{M}', s'_1) & \models \left(\bigwedge_{(s_1,s_2) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) & \wedge \\
& \text{AX} \left(\bigvee_{(s_1,s_2) \in R} \mathcal{F}_V(\text{Tr}_c(s_2)) \right) & \text{((2), (5))} \\
\text{(7)} \quad \dots & \\
\text{(8)} \quad (\mathcal{M}', s'_k) & \models \left(\bigwedge_{(s_k,s_{k+1}) \in R} \text{EX} \mathcal{F}_V(\text{Tr}_c(s_{k+1})) \right) & \wedge \\
& \text{AX} \left(\bigvee_{(s_k,s_{k+1}) \in R} \mathcal{F}_V(\text{Tr}_c(s_{k+1})) \right) & \text{(similar with (6))} \\
\text{(9)} \quad \forall (s_k, s_{k+1}) \in R, \exists (s'_k, s'_{k+1}) \in R' & (\mathcal{M}', s'_{k+1}) \models \mathcal{F}_V(\text{Tr}_c(s_{k+1})) & \text{(8)} \\
\text{(10)} \quad \text{Tr}_c(s_{k+1}) \leftrightarrow_{\bar{V}} \text{Tr}_c(s'_{k+1}) & \text{((9), Lemma 6)} \\
\text{(11)} \quad L(s_{k+1}) - \bar{V} = L'(s'_{k+1}) - \bar{V} & \text{((10), } c \geq 0)
\end{aligned}$$

(c) We will show that for each $(s'_k, s'_{k+1}) \in R'$ there is a $(s_k, s_{k+1}) \in R$ such that $L(s_{k+1}) - \bar{V} = L'(s'_{k+1}) - \bar{V}$.

$$\begin{aligned}
\text{(1)} \quad (\mathcal{M}', s'_k) & \models \text{AX} \left(\bigvee_{(s_k,s_{k+1}) \in R} \mathcal{F}_V(\text{Tr}_c(s_{k+1})) \right) & \text{(by (8) talked above)} \\
\text{(2)} \quad \forall (s'_k, s'_{k+1}) \in R', \exists (s_k, s_{k+1}) \in R & (\mathcal{M}', s'_{k+1}) \models \mathcal{F}_V(\text{Tr}_c(s_{k+1})) & \text{(1)} \\
\text{(3)} \quad \text{Tr}_c(s_{k+1}) \leftrightarrow_{\bar{V}} \text{Tr}_c(s'_{k+1}) & \text{((2), Lemma 6)} \\
\text{(4)} \quad L(s_{k+1}) - \bar{V} = L'(s'_{k+1}) - \bar{V} & \text{((3), } c \geq 0)
\end{aligned}$$

(ii) This is following Lemma 2 and the definition of the characterizing formula of initial K-structure \mathcal{K} on V . \blacksquare

Lemma 3 Let φ be a formula. We have

$$\varphi \equiv \bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}}(\mathcal{M}, s_0). \quad (4)$$

Proof: Let (\mathcal{M}', s'_0) be a model of φ . Then $(\mathcal{M}', s'_0) \models \bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}}(\mathcal{M}, s_0)$ due to $(\mathcal{M}', s'_0) \models \mathcal{F}_{\mathcal{A}}(\mathcal{M}', s'_0)$. On the other hand, suppose that (\mathcal{M}', s'_0) is a model of $\bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}}(\mathcal{M}, s_0)$.

Then there is a $(\mathcal{M}, s_0) \in \text{Mod}(\varphi)$ such that $(\mathcal{M}', s'_0) \models \mathcal{F}_{\mathcal{A}}(\mathcal{M}, s_0)$. And then $(\mathcal{M}, s_0) \leftrightarrow_{\varnothing} (\mathcal{M}', s'_0)$ by Theorem 2. Therefore, (\mathcal{M}, s_0) is also a model of φ by Theorem 1. \blacksquare

Theorem 3 (Representation theorem) Let φ, φ' and ϕ be CTL formulas and $V \subseteq \mathcal{A}$. Then the following statements are equivalent:

- (i) $\varphi' \equiv \text{F}_{\text{CTL}}(\varphi, V)$,
- (ii) $\varphi' \equiv \{\phi \mid \varphi \models \phi \text{ and } \text{IR}(\phi, V)\}$,
- (iii) Postulates (W), (PP), (NP) and (IR) hold.

Proof: (i) \Leftrightarrow (ii). To prove this, we will show that:

$$\begin{aligned}
& \text{Mod}(\text{F}_{\text{CTL}}(\varphi, V)) = \text{Mod}(\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\}) \\
& = \text{Mod}\left(\bigvee_{\mathcal{M}, s_0 \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}-V}(\mathcal{M}, s_0)\right).
\end{aligned}$$

Firstly, suppose that (\mathcal{M}', s'_0) is a model of $\text{F}_{\text{CTL}}(\varphi, V)$. Then there exists an initial K-structure (\mathcal{M}, s_0) such that (\mathcal{M}, s_0) is a model of φ and $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$. By Theorem 1, we have $(\mathcal{M}', s'_0) \models \phi$ for all ϕ that $\varphi \models \phi$ and $\text{IR}(\phi, V)$. Thus, (\mathcal{M}', s'_0) is a model of $\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\}$.

Secondly, suppose that (\mathcal{M}', s'_0) is a model of $\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\}$. Thus, $(\mathcal{M}', s'_0) \models \bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}-V}(\mathcal{M}, s_0)$ due to $\bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}-V}(\mathcal{M}, s_0)$ is irrelevant to V .

Finally, suppose that (\mathcal{M}', s'_0) is a model of $\bigvee_{(\mathcal{M}, s_0) \in \text{Mod}(\varphi)} \mathcal{F}_{\mathcal{A}-V}(\mathcal{M}, s_0)$. Then there exists $(\mathcal{M}, s_0) \in \text{Mod}(\varphi)$ such that $(\mathcal{M}', s'_0) \models \mathcal{F}_{\mathcal{A}-V}(\mathcal{M}, s_0)$. Hence, $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$ by Theorem 2. Thus (\mathcal{M}', s'_0) is also a model of $\text{F}_{\text{CTL}}(\varphi, V)$.

(ii) \Rightarrow (iii). It is not difficult to prove it.

(iii) \Rightarrow (ii). Suppose that all postulates hold. By Positive Persistence, we have $\varphi' \models \{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\}$. Now we show that $\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\} \models \varphi'$. Otherwise, there exists formula ϕ' such that $\varphi' \models \phi'$ but $\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\} \not\models \phi'$. There are three cases:

- ϕ' is relevant to V . Thus, φ' is also relevant to V , a contradiction to Irrelevance.
- ϕ' is irrelevant to V and $\varphi \models \phi'$. This contradicts to our assumption.
- ϕ' is irrelevant to V and $\varphi \not\models \phi'$. By Negative Persistence, $\varphi' \not\models \phi'$, a contradiction.

Thus, φ' is equivalent to $\{\phi \mid \varphi \models \phi, \text{IR}(\phi, V)\}$. \blacksquare

Lemma 4 Let φ and α be two CTL formulae and $q \in \overline{\text{Var}(\varphi) \cup \text{Var}(\alpha)}$. Then $\text{F}_{\text{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q) \equiv \varphi$.

Proof: Let $\varphi' = \varphi \wedge (q \leftrightarrow \alpha)$. For any model (\mathcal{M}, s) of $\text{F}_{\text{CTL}}(\Gamma', q)$ there is an initial K-structure (\mathcal{M}', s') s.t. $(\mathcal{M}, s) \leftrightarrow_{\{q\}} (\mathcal{M}', s')$ and $(\mathcal{M}', s') \models \varphi'$. It's apparent that $(\mathcal{M}', s') \models \varphi$, and then $(\mathcal{M}, s) \models \varphi$ since $\text{IR}(\varphi, \{q\})$ and $(\mathcal{M}, s) \leftrightarrow_{\{q\}} (\mathcal{M}', s')$ by Theorem 1.

Let $(\mathcal{M}, s) \in \text{Mod}(\varphi)$ with $\mathcal{M} = (S, R, L, s)$. We construct (\mathcal{M}', s) with $\mathcal{M}' = (S, R, L', s)$ as follows:

$L' : S \rightarrow \mathcal{A}$ and $\forall s^* \in S, L'(s^*) = L(s^*)$ if $(\mathcal{M}, s^*) \models \alpha$,
else $L'(s^*) = L(s^*) \cup \{q\}$,

$L'(s) = L(s) \cup \{q\}$ if $(\mathcal{M}, s) \models \alpha$, and $L'(s) = L(s)$

otherwise.

It is clear that $(\mathcal{M}', s) \models \varphi$, $(\mathcal{M}', s) \models q \leftrightarrow \alpha$ and $(\mathcal{M}', s) \leftrightarrow_{\{q\}} (\mathcal{M}, s)$. Therefore $(\mathcal{M}', s) \models \varphi \wedge (q \leftrightarrow \alpha)$, and then $(\mathcal{M}, s) \models \text{F}_{\text{CTL}}(\varphi \wedge (q \leftrightarrow \alpha), q)$ by $(\mathcal{M}', s) \leftrightarrow_{\{q\}} (\mathcal{M}, s)$. ■

Proposition 4 Let φ be a formula, V a set of atoms and p an atom such that $p \notin V$. Then:

$$\text{F}_{\text{CTL}}(\varphi, \{p\} \cup V) \equiv \text{F}_{\text{CTL}}(\text{F}_{\text{CTL}}(\varphi, p), V).$$

Proof: Let (\mathcal{M}_1, s_1) with $\mathcal{M}_1 = (S_1, R_1, L_1, s_1)$ be a model of $\text{F}_{\text{CTL}}(\varphi, \{p\} \cup V)$. By the definition, there exists a model (\mathcal{M}, s) with $\mathcal{M} = (S, R, L, s)$ of φ , such that $(\mathcal{M}_1, s_1) \leftrightarrow_{\{p\} \cup V} (\mathcal{M}, s)$. We construct an initial K-structure (\mathcal{M}_2, s_2) with $\mathcal{M}_2 = (S_2, R_2, L_2, s_2)$ as follows:

(1) for s_2 : let s_2 be the state such that:

- $p \in L_2(s_2)$ iff $p \in L_1(s_1)$,
- for all $q \in V, q \in L_2(s_2)$ iff $q \in L(s)$,
- for all other atoms $q', q' \in L_2(s_2)$ iff $q' \in L_1(s_1)$ iff $q' \in L(s)$.

(2) for another:

- (i) for all pairs $w \in S$ and $w_1 \in S_1$ such that $w \leftrightarrow_{\{p\} \cup V} w_1$, let $w_2 \in S_2$ and
 - $p \in L_2(w_2)$ iff $p \in L_1(w_1)$,
 - for all $q \in V, q \in L_2(w_2)$ iff $q \in L(w)$,
 - for all other atoms $q', q' \in L_2(w_2)$ iff $q' \in L_1(w_1)$ iff $q' \in L(w)$.
- (ii) if $(w'_1, w_1) \in R_1$, w_2 is constructed based on w_1 and $w'_2 \in S_2$ is constructed based on w'_1 , then $(w'_2, w_2) \in R_2$.

(3) delete duplicated states in S_2 and pairs in R_2 .

Then we have $(\mathcal{M}, s) \leftrightarrow_{\{p\}} (\mathcal{M}_2, s_2)$ and $(\mathcal{M}_2, s_2) \leftrightarrow_V (\mathcal{M}_1, s_1)$. Thus, $(\mathcal{M}_2, s_2) \models \text{F}_{\text{CTL}}(\varphi, p)$. And therefore $(\mathcal{M}_1, s_1) \models \text{F}_{\text{CTL}}(\text{F}_{\text{CTL}}(\varphi, p), V)$.

On the other hand, suppose that (\mathcal{M}_1, s_1) be a model of $\text{F}_{\text{CTL}}(\text{F}_{\text{CTL}}(\varphi, p), V)$, then there exists an initial K-structure (\mathcal{M}_2, s_2) such that $(\mathcal{M}_2, s_2) \models \text{F}_{\text{CTL}}(\varphi, p)$ and $(\mathcal{M}_2, s_2) \leftrightarrow_V (\mathcal{M}_1, s_1)$, and there exists (\mathcal{M}, s) such that $(\mathcal{M}, s) \models \varphi$ and $(\mathcal{M}, s) \leftrightarrow_{\{p\}} (\mathcal{M}_2, s_2)$. Therefore, $(\mathcal{M}, s) \leftrightarrow_{\{p\} \cup V} (\mathcal{M}_1, s_1)$ by Proposition 1, and consequently, $(\mathcal{M}_1, s_1) \models \text{F}_{\text{CTL}}(\varphi, \{p\} \cup V)$. ■

Proposition 5 Let $\varphi, \varphi_i, \psi_i$ ($i = 1, 2$) be formulas and $V \subseteq \mathcal{A}$. We have

- (i) $\text{F}_{\text{CTL}}(\varphi, V)$ is satisfiable iff φ is;
- (ii) If $\varphi_1 \equiv \varphi_2$, then $\text{F}_{\text{CTL}}(\varphi_1, V) \equiv \text{F}_{\text{CTL}}(\varphi_2, V)$;
- (iii) If $\varphi_1 \models \varphi_2$, then $\text{F}_{\text{CTL}}(\varphi_1, V) \models \text{F}_{\text{CTL}}(\varphi_2, V)$;

(iv) $\text{F}_{\text{CTL}}(\psi_1 \vee \psi_2, V) \equiv \text{F}_{\text{CTL}}(\psi_1, V) \vee \text{F}_{\text{CTL}}(\psi_2, V)$;

(v) $\text{F}_{\text{CTL}}(\psi_1 \wedge \psi_2, V) \models \text{F}_{\text{CTL}}(\psi_1, V) \wedge \text{F}_{\text{CTL}}(\psi_2, V)$;

Proof: (i) (\Rightarrow) Supposing (\mathcal{M}, s) is a model of $\text{F}_{\text{CTL}}(\varphi, V)$, then there is a model (\mathcal{M}', s') of φ s.t. $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}', s')$ by the definition of F_{CTL} .

(\Leftarrow) Supposing (\mathcal{M}, s) is a model of φ , then there is an initial Kripke structure (\mathcal{M}', s') s.t. $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}', s')$, and then $(\mathcal{M}', s') \models \text{F}_{\text{CTL}}(\varphi, V)$ by the definition of F_{CTL} .

The (ii) and (iii) can be proved similarly.

(iv) $(\Rightarrow) \forall (\mathcal{M}, s) \in \text{Mod}(\text{F}_{\text{CTL}}(\psi_1 \vee \psi_2, V)), \exists (\mathcal{M}', s') \in \text{Mod}(\psi_1 \vee \psi_2)$ s.t. $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}', s')$ and $(\mathcal{M}', s') \models \psi_1$ or $(\mathcal{M}', s') \models \psi_2$

$\Rightarrow \exists (\mathcal{M}_1, s_1) \in \text{Mod}(\text{F}_{\text{CTL}}(\psi_1, V))$ s.t. $(\mathcal{M}', s') \leftrightarrow_V (\mathcal{M}_1, s_1)$ or $\exists (\mathcal{M}_2, s_2) \in \text{Mod}(\text{F}_{\text{CTL}}(\psi_2, V))$ s.t. $(\mathcal{M}', s') \leftrightarrow_V (\mathcal{M}_2, s_2)$

$\Rightarrow (\mathcal{M}, s) \models \text{F}_{\text{CTL}}(\psi_1, V) \vee \text{F}_{\text{CTL}}(\psi_2, V)$ by Theorem 1.

$(\Leftarrow) \forall (\mathcal{M}, s) \in \text{Mod}(\text{F}_{\text{CTL}}(\psi_1, V) \vee \text{F}_{\text{CTL}}(\psi_2, V))$

$\Rightarrow (\mathcal{M}, s) \models \text{F}_{\text{CTL}}(\psi_1, V)$ or $(\mathcal{M}, s) \models \text{F}_{\text{CTL}}(\psi_2, V)$

\Rightarrow there is an initial K-structure (\mathcal{M}_1, s_1) s.t. $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_1, s_1)$ and $(\mathcal{M}_1, s_1) \models \psi_1$ or $(\mathcal{M}_1, s_1) \models \psi_2$

$\Rightarrow (\mathcal{M}_1, s_1) \models \psi_1 \vee \psi_2$

\Rightarrow there is an initial K-structure (\mathcal{M}_2, s_2) s.t. $(\mathcal{M}_1, s_1) \leftrightarrow_V (\mathcal{M}_2, s_2)$ and $(\mathcal{M}_2, s_2) \models \text{F}_{\text{CTL}}(\psi_1 \vee \psi_2, V)$

$\Rightarrow (\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_2, s_2)$ and $(\mathcal{M}, s) \models \text{F}_{\text{CTL}}(\psi_1 \vee \psi_2, V)$.

The (v) can be proved as (iv). ■

Proposition 6 (Homogeneity) Let $V \subseteq \mathcal{A}$ and $\phi \in \text{CTL}$,

(i) $\text{F}_{\text{CTL}}(\text{AX}\phi, V) \equiv \text{AXF}_{\text{CTL}}(\phi, V)$.

(ii) $\text{F}_{\text{CTL}}(\text{EX}\phi, V) \equiv \text{EXF}_{\text{CTL}}(\phi, V)$.

(iii) $\text{F}_{\text{CTL}}(\text{AF}\phi, V) \equiv \text{AFF}_{\text{CTL}}(\phi, V)$.

(iv) $\text{F}_{\text{CTL}}(\text{EF}\phi, V) \equiv \text{EFF}_{\text{CTL}}(\phi, V)$.

Proof: Let $\mathcal{M} = (S, R, L, s_0)$ with initial state s_0 and $\mathcal{M}' = (S', R', L', s'_0)$ with initial state s'_0 , then we call \mathcal{M}', s'_0 be a sub-structure of \mathcal{M}, s_0 if:

- $S' \subseteq S$ and $S' = \{s' | s' \text{ is reachable from } s'_0\}$,
- $R' = \{(s_1, s_2) | s_1, s_2 \in S' \text{ and } (s_1, s_2) \in R\}$,
- $L' : S' \rightarrow \mathcal{A}$ and $\forall s_1 \in S'$ there is $L'(s_1) = L(s_1)$, and
- s'_0 is s_0 or a state reachable from s_0 .

(i) In order to prove $\text{F}_{\text{CTL}}(\text{AX}\phi, V) \equiv \text{AX}(\text{F}_{\text{CTL}}(\phi, V))$, we only need to prove $\text{Mod}(\text{F}_{\text{CTL}}(\text{AX}\phi, V)) = \text{Mod}(\text{AXF}_{\text{CTL}}(\phi, V))$:

$(\Rightarrow) \forall (\mathcal{M}', s') \in \text{Mod}(\text{F}_{\text{CTL}}(\text{AX}\phi, V))$ there exists an initial K-structure (\mathcal{M}, s) s.t. $(\mathcal{M}, s) \models \text{AX}\phi$ and $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}', s')$

\Rightarrow for any sub-structure (\mathcal{M}_1, s_1) of (\mathcal{M}, s) there is $(\mathcal{M}_1, s_1) \models \phi$, where s_1 is a directed successor of s

\Rightarrow there is an initial K-structure (\mathcal{M}_2, s_2) s.t. $(\mathcal{M}_2, s_2) \models \text{F}_{\text{CTL}}(\phi, V)$ and $(\mathcal{M}_2, s_2) \leftrightarrow_V (\mathcal{M}_1, s_1)$

\Rightarrow it is easy to construct an initial K-structure (\mathcal{M}_3, s_3) by (\mathcal{M}_2, s_2) s.t. (\mathcal{M}_2, s_2) is a sub-structure of (\mathcal{M}_3, s_3) with s_2 is a direct successor of s_3 and $(\mathcal{M}_3, s_3) \leftrightarrow_V (\mathcal{M}, s)$

$\Rightarrow (\mathcal{M}_3, s_3) \models \text{AX}(\text{F}_{\text{CTL}}(\phi, V))$ and $(\mathcal{M}_3, s_3) \leftrightarrow_V (\mathcal{M}', s')$

$\Rightarrow (\mathcal{M}', s') \models \text{AX}(\text{F}_{\text{CTL}}(\phi, V))$.

$(\Leftarrow) \forall (\mathcal{M}_3, s_3) \in \text{Mod}(\text{AX}(\text{F}_{\text{CTL}}(\phi, V)))$, then for any sub-structure (\mathcal{M}_2, s_2) with s_2 is a directed successor of s_3

there is $(\mathcal{M}_2, s_2) \models F_{CTL}(\phi, V)$
 \Rightarrow for any (\mathcal{M}_2, s_2) there is an initial K-structure (\mathcal{M}_1, s_1)
s.t. $(\mathcal{M}_1, s_1) \models \phi$ and $(\mathcal{M}_1, s_1) \leftrightarrow_V (\mathcal{M}_2, s_2)$
 \Rightarrow it is easy to construct an initial structure (\mathcal{M}, s) by
 (\mathcal{M}_1, s_1) s.t. (\mathcal{M}_1, s_1) is a sub-structure of (\mathcal{M}, s) with s_1
is a direct successor of s and $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_3, s_3)$
 $\Rightarrow (\mathcal{M}, s) \models AX\phi$ and then $(\mathcal{M}_3, s_3) \models F_{CTL}(AX\phi, V)$.

(ii) In order to prove $F_{CTL}(EX\phi, V) \equiv EXF_{CTL}(\phi, V)$,
we only need to prove $Mod(F_{CTL}(EX\phi, V)) =$
 $Mod(EXF_{CTL}(\phi, V))$:

$(\Rightarrow) \forall \mathcal{M}', s' \in Mod(F_{CTL}(EX\phi, V))$ there exists an ini-
tial K-structure (\mathcal{M}, s) s.t. $(\mathcal{M}, s) \models EX\phi$ and $(\mathcal{M}, s) \leftrightarrow_V$
 (\mathcal{M}', s')

\Rightarrow there is a sub-structure (\mathcal{M}_1, s_1) of (\mathcal{M}, s) s.t.
 $(\mathcal{M}_1, s_1) \models \phi$, where s_1 is a directed successor of s

\Rightarrow there is an initial K-structure (\mathcal{M}_2, s_2) s.t. $(\mathcal{M}_2, s_2) \models$
 $F_{CTL}(\phi, V)$ and $(\mathcal{M}_2, s_2) \leftrightarrow_V (\mathcal{M}_1, s_1)$

\Rightarrow it is easy to construct an initial K-structure (\mathcal{M}_3, s_3) by
 (\mathcal{M}_2, s_2) s.t. (\mathcal{M}_2, s_2) is a sub-structure of (\mathcal{M}_3, s_3) that s_2
is a direct successor of s_3 and $(\mathcal{M}_3, s_3) \leftrightarrow_V (\mathcal{M}, s)$

$\Rightarrow (\mathcal{M}_3, s_3) \models EX(F_{CTL}(\phi, V))$

$\Rightarrow (\mathcal{M}', s') \models EX(F_{CTL}(\phi, V))$.

$(\Leftarrow) \forall (\mathcal{M}_3, s_3) \in Mod(EX(F_{CTL}(\phi, V)))$, then there ex-
ists a sub-structure (\mathcal{M}_2, s_2) of (\mathcal{M}_3, s_3) s.t. $(\mathcal{M}_2, s_2) \models$
 $F_{CTL}(\phi, V)$

\Rightarrow there is an initial K-structure (\mathcal{M}_1, s_1) s.t. $(\mathcal{M}_1, s_1) \models \phi$
and $(\mathcal{M}_1, s_1) \leftrightarrow_V (\mathcal{M}_2, s_2)$

\Rightarrow it is easy to construct an initial K-structure (\mathcal{M}, s) by
 (\mathcal{M}_1, s_1) s.t. (\mathcal{M}_1, s_1) is a sub-structure of (\mathcal{M}, s) that s_1
is a direct successor of s and $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_3, s_3)$

$\Rightarrow (\mathcal{M}, s) \models EX\phi$ and then $(\mathcal{M}_3, s_3) \models F_{CTL}(EX\phi, V)$.

(iii) and (iv) can be proved as (i) and (ii) respectively. ■

Proposition 7 (Model Checking on Forgetting) Let
 (\mathcal{M}, s_0) be an initial K-structure, φ be a CTL formula and
 V a set of atoms. Deciding whether (\mathcal{M}, s_0) is a model of
 $F_{CTL}(\varphi, V)$ is NP-complete.

Proof: The problem can be determined by the following two
things: (1) guessing an initial K-structure (\mathcal{M}', s'_0) satisfying
 φ ; and (2) checking if $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$. Both two
steps can be done in polynomial time. Hence, the problem
is in NP. The hardness follows that the model checking for
propositional variable forgetting is NP-hard [Zhang and
Zhou, 2008]. ■

Theorem 5 (Entailment on Forgetting) Let φ and ψ be
two CTL_{AF} formulas and V a set of atoms. Then, results:

- (i) deciding $F_{CTL}(\varphi, V) \models^? \psi$ is co-NP-complete,
- (ii) deciding $\psi \models^? F_{CTL}(\varphi, V)$ is Π_2^P -complete,
- (iii) deciding $F_{CTL}(\varphi, V) \models^? F_{CTL}(\psi, V)$ is Π_2^P -complete.

Proof: (1) It is proved that deciding whether ψ is satis-
fiable is NP-Complete [Meier *et al.*, 2015]. The hardness is
easy to see by setting $F_{CTL}(\varphi, Var(\varphi)) \equiv \top$, i.e., deciding
whether ψ is valid. For membership, from Theorem 3, we
have $F_{CTL}(\varphi, V) \models \psi$ iff $\varphi \models \psi$ and $IR(\psi, V)$. Clearly, in
 CTL_{AF} , deciding $\varphi \models \psi$ is in co-NP. We show that deciding
whether $IR(\psi, V)$ is also in co-NP. Without loss of gener-
ality, we assume that ψ is satisfiable. We consider the comple-

ment of the problem: deciding whether ψ is not irrelevant to
 V . It is easy to see that ψ is not irrelevant to V iff there exist a
model (\mathcal{M}, s_0) of ψ and an initial K-structure (\mathcal{M}', s'_0) such
that $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$ and $(\mathcal{M}', s'_0) \not\models \psi$. So checking
whether ψ is not irrelevant to V can be achieved in the fol-
lowing steps: (1) guess two initial K-structures (\mathcal{M}, s_0) and
 (\mathcal{M}', s'_0) , (2) check if $(\mathcal{M}, s_0) \models \psi$ and $(\mathcal{M}', s'_0) \not\models \psi$, and
(3) check $(\mathcal{M}, s_0) \leftrightarrow_V (\mathcal{M}', s'_0)$. Obviously (1) can be done
in polynomial time and also (2) and (3) can be done in poly-
nomial time.

(2) Membership. We consider the complement of the prob-
lem. We may guess an initial K-structure (\mathcal{M}, s_0) and check
whether $(\mathcal{M}, s_0) \models \psi$ and $(\mathcal{M}, s_0) \not\models F_{CTL}(\varphi, V)$. From
Proposition 7, we know that this is in Σ_2^P . So the original
problem is in Π_2^P . Hardness. Let $\psi \equiv \top$. Then the problem
is reduced to decide $F_{CTL}(\varphi, V)$'s validity. Since a proposi-
tional variable forgetting is a special case temporal forgetting,
the hardness is directly followed from the proof of Proposi-
tion 24 in [Lang *et al.*, 2003].

(3) Membership. If $F_{CTL}(\varphi, V) \not\models F_{CTL}(\psi, V)$ then there
exist an initial K-structure (\mathcal{M}, s) such that $(\mathcal{M}, s) \models$
 $F_{CTL}(\varphi, V)$ but $(\mathcal{M}, s) \not\models F_{CTL}(\psi, V)$, i.e., there is
 $(\mathcal{M}_1, s_1) \leftrightarrow_V (\mathcal{M}, s)$ with $(\mathcal{M}_1, s_1) \models \varphi$ but $(\mathcal{M}_2, s_2) \not\models \psi$
for every (\mathcal{M}_2, s_2) with $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_2, s_2)$. It is evi-
dent that guessing such (\mathcal{M}, s) , (\mathcal{M}_1, s_1) with $(\mathcal{M}_1, s_1) \leftrightarrow_V$
 (\mathcal{M}, s) and checking $(\mathcal{M}_1, s_1) \models \varphi$ are feasible while check-
ing $(\mathcal{M}_2, s_2) \not\models \psi$ for every $(\mathcal{M}, s) \leftrightarrow_V (\mathcal{M}_2, s_2)$ can be
done in polynomial time. Thus the problem is in Π_2^P .

Hardness. It follows from (2) due to the fact that
 $F_{CTL}(\varphi, V) \models F_{CTL}(\psi, V)$ iff $\varphi \models F_{CTL}(\psi, V)$ thanks to
 $IR(F_{CTL}(\psi, V), V)$. ■

Proposition 8 (dual) Let V, q, φ and ψ are the ones in Def-
inition 4. The ψ is a SNC (WSC) of q on V under φ iff $\neg\psi$ is
a WSC (SNC) of $\neg q$ on V under φ .

Proof: (i) Suppose ψ is the SNC of q . Then $\varphi \models q \rightarrow \psi$.
Thus $\varphi \models \neg\psi \rightarrow \neg q$. So $\neg\psi$ is a SC of $\neg q$. Suppose ψ' is
any other SC of $\neg q$: $\varphi \models \psi' \rightarrow \neg q$. Then $\varphi \models q \rightarrow \neg\psi'$, this
means $\neg\psi'$ is a NC of q on P under φ . Thus $\varphi \models \psi \rightarrow \neg\psi'$
by assumption. So $\varphi \models \psi' \rightarrow \neg\psi$. This proves that $\neg\psi$ is the
WSC of $\neg q$. The proof of the other part of the proposition is
similar.

(ii) The WSC case can be proved similarly with SNC case. ■

Proposition 9 Let Γ and α be two formulas, $V \subseteq Var(\alpha) \cup$
 $Var(\phi)$ and q is a new proposition not in Γ and α . Then, a
formula φ of V is the SNC (WSC) of α on V under Γ iff it is
the SNC (WSC) of q on V under $\Gamma' = \Gamma \cup \{q \equiv \alpha\}$.

Proof: We prove this for SNC. The case for WSC is simi-
lar. Let $SNC(\varphi, \alpha, V, \Gamma)$ denote that φ is the SNC of α on V
under Γ , and $NC(\varphi, \alpha, V, \Gamma)$ denote that φ is the NC of α on
 V under Γ .

(\Rightarrow) We will show that if $SNC(\varphi, \alpha, V, \Gamma)$ holds, then
 $SNC(\varphi, q, V, \Gamma')$ will be true. According to $SNC(\varphi, \alpha, V, \Gamma)$
and $\alpha \equiv q$, we have $\Gamma' \models q \rightarrow \varphi$, which means φ is a NC of
 q on V under Γ' . Suppose φ' is any NC of q on V under Γ' ,
then $F_{CTL}(\Gamma', q) \models \alpha \rightarrow \varphi'$ due to $\alpha \equiv q$, $IR(\alpha \rightarrow \varphi', \{q\})$

and (PP), i.e., $\Gamma \models \alpha \rightarrow \varphi'$ by Lemma 4, this means $NC(\varphi', \alpha, V, \Gamma)$. Therefore, $\Gamma \models \varphi \rightarrow \varphi'$ by the definition of SNC and $\Gamma' \models \varphi \rightarrow \varphi'$. Hence, $SNC(\varphi, q, V, \Gamma')$ holds.

(\Leftarrow) We will show that if $SNC(\varphi, q, V, \Gamma')$ holds, then $SNC(\varphi, \alpha, V, \Gamma)$ will be true. According to $SNC(\varphi, q, V, \Gamma')$, it's not difficult to know that $F_{CTL}(\Gamma', \{q\}) \models \alpha \rightarrow \varphi$ due to $\alpha \equiv q$, $IR(\alpha \rightarrow \varphi, \{q\})$ and (PP), i.e., $\Gamma \models \alpha \rightarrow \varphi$ by Lemma 4, this means $NC(\varphi, \alpha, V, \Gamma)$. Suppose φ' is any NC of α on V under Γ . Then $\Gamma' \models q \rightarrow \varphi'$ since $\alpha \equiv q$ and $\Gamma' = \Gamma \cup \{q \equiv \alpha\}$, which means $NC(\varphi', q, V, \Gamma')$. According to $SNC(\varphi, q, V, \Gamma')$, $IR(\varphi \rightarrow \varphi', \{q\})$ and (PP), we have $F_{CTL}(\Gamma', \{q\}) \models \varphi \rightarrow \varphi'$, and $\Gamma \models \varphi \rightarrow \varphi'$ by Lemma 4. Hence, $SNC(\varphi, \alpha, V, \Gamma)$ holds. ■

Theorem 7 Let φ be a formula, $V \subseteq Var(\varphi)$ and $q \in Var(\varphi) - V$.

- (i) $F_{CTL}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$ is a SNC of q on V under φ .
- (ii) $\neg F_{CTL}(\varphi \wedge \neg q, (Var(\varphi) \cup \{q\}) - V)$ is a WSC of q on V under φ .

Proof: We will prove the SNC part, while it is not difficult to prove the WSC part according to Proposition 8. Let $\mathcal{F} = F_{CTL}(\varphi \wedge q, (Var(\varphi) \cup \{q\}) - V)$.

The “NC” part: It's easy to see that $\varphi \wedge q \models \mathcal{F}$ by (W). Hence, $\varphi \models q \rightarrow \mathcal{F}$, this means \mathcal{F} is a NC of q on P under φ .

The “SNC” part: for all ψ', ψ' is the NC of q on V under φ , s.t. $\varphi \models \mathcal{F} \rightarrow \psi'$. Suppose that there is a NC ψ of q on V under φ and ψ is not logic equivalence with \mathcal{F} under φ , s.t. $\varphi \models \psi \rightarrow \mathcal{F}$. We know that $\varphi \wedge q \models \psi$ iff $\mathcal{F} \models \psi$ by (PP), since $IR(\psi, (Var(\varphi) \cup \{q\}) - V)$. Hence, $\varphi \wedge \mathcal{F} \models \psi$ by $\varphi \wedge q \models \psi$ (by suppose). We can see that $\varphi \wedge \psi \models \mathcal{F}$ by suppose. Therefore, $\varphi \models \psi \leftrightarrow \mathcal{F}$, which means ψ is logic equivalence with \mathcal{F} under φ . This is contradict with the suppose. Then \mathcal{F} is the SNC of q on P under φ . ■

Theorem 8 Let $\mathcal{K} = (\mathcal{M}, s)$ be an initial K-structure with $\mathcal{M} = (S, R, L, s_0)$ on the finite set \mathcal{A} of atoms, $V \subseteq \mathcal{A}$ and $q \in V'$ ($V' = \mathcal{A} - V$). Then:

- (i) the SNC of q on V under \mathcal{K} is $F_{CTL}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge q, V')$.
- (ii) the WSC of q on V under \mathcal{K} is $\neg F_{CTL}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge \neg q, V')$.

Proof: (i) As we know that any initial K-structure \mathcal{K} can be described as a characterizing formula $\mathcal{F}_{\mathcal{A}}(\mathcal{K})$, then the SNC of q on V under $\mathcal{F}_{\mathcal{A}}(\mathcal{K})$ is $F_{CTL}(\mathcal{F}_{\mathcal{A}}(\mathcal{K}) \wedge q, \mathcal{A} - V)$.

(ii) This is proved by the dual property. ■

Proposition 10 Let φ be a CTL formula and $V \subseteq \mathcal{A}$ with $|\mathcal{A}| = m$ and $|V| = n$. The time and space complexity of Algorithm 1 are $O(2^{m*2^m})$.

Proof: The time and space spent by Algorithm 1 is mainly the for cycles between lines 4 and 18. Under a given number i of states, there are i^i number of relations, i^{2^m} number of label functions and i number of possible initial states. In this case, we need the memory for the initial K-model in each time is $(i + i^i + i^{2^m} + 1)$.

For each $1 \leq i \leq 2^m$, there is at most $i * i^i * i^{2^m} * i = i^2 * i^{(i+2^m)}$ possible initial K-models. Suppose that we can

obtain an initial K-models in unit time (at each step), then we require $(2^m)^2 * (2^m)^{(2^m+2^m)} = (2^m)^{(2+2*2^m)}$ steps in the worst case.

Let $k = m - n$, for any initial K-structure $\mathcal{K} = (\mathcal{M}, s_0)$ with $i \geq 1$ nodes, in the worst, i.e., $ch(\mathcal{M}, V) = i$, we will spend $N(i)$ space to store the characterizing formula.

$$\begin{aligned} N(i) &= (k + (\dots + (k + 2ik) * (2i)) \dots * (2i)) \\ &= (2i)^0 k + 2ik + (2i)^2 k + \dots + (2i)^{(i-1)} k \\ &= \frac{(2i)^i - 1}{2i - 1} k. \end{aligned}$$

In the worst case, i.e., there is $i = 2^m$ initial K-structure with 2^m nodes, we will spent $2^m * N(i)$ space to store the result of forgetting.

It is obvious that computing the V -characterization number of any initial K-structure \mathcal{K} dose not more than $O(i^2 P)$ with P expressing a polynomial function. Therefore, the time and space complexity are $O(2^{(m*2^m)})$. ■

References

- [Baier and Katoen, 2008] Christel Baier and JoostPieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.
- [Bolotov, 1999] Alexander Bolotov. A clausal resolution method for ctl branching-time temporal logic. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(1):77–93, 1999.
- [Browne et al., 1988] Michael C. Browne, Edmund M. Clarke, and Orna Grumberg. Characterizing finite kripke structures in propositional temporal logic. *Theor. Comput. Sci.*, 59:115–131, 1988.
- [Clarke and Emerson, 1981] Edmund M Clarke and E Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on Logic of Programs*, pages 52–71. Springer, 1981.
- [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 Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.
- [Dailier et al., 2018] Sylvain Dailier, David Hauzar, Claude Marché, and Yannick Moy. Instrumenting a weakest precondition calculus for counterexample generation. *Journal of logical and algebraic methods in programming*, 99:97–113, 2018.
- [Dijkstra, 1978] Edsger W Dijkstra. Guarded commands, nondeterminacy, and formal derivation of programs. In *Programming Methodology*, pages 166–175. Springer, 1978.
- [Ding and Zhang, 2006] Yulin Ding and Yan Zhang. Ctl model update: Semantics, computations and implementation. In *ECAI*, pages 362–366, 2006.
- [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.
- [Lang and Marquis, 2008] Jerome Lang and Pierre Marquis. On propositional definability. *Artificial Intelligence*, 172(8):991–1017, 2008.
- [Lang and Marquis, 2010] Jerome Lang and Pierre Marquis. *Reasoning under inconsistency: a forgetting-based approach*. Elsevier Science Publishers Ltd, 2010.
- [Lang *et al.*, 2003] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional independence: Formula-variable independence and forgetting. *J. Artif. Intell. Res.*, 18:391–443, 2003.
- [Lin and Reiter, 1994] Fangzhen Lin and Ray Reiter. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.
- [Lin, 2001] Fangzhen Lin. On strongest necessary and weakest sufficient conditions. *Artif. Intell.*, 128(1-2):143–159, 2001.
- [Lin, 2003] Fangzhen Lin. Compiling causal theories to successor state axioms and strips-like systems. *Journal of Artificial Intelligence Research*, 19:279–314, 2003.
- [Lin, 2018] Fangzhen Lin. Machine theorem discovery. *AI Magazine*, 39(2):53–59, 2018.
- [Liu and Wen, 2011] Yongmei Liu and Ximing Wen. On the progression of knowledge in the situation calculus. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 976–982, Barcelona, Catalonia, Spain, 2011. IJCAI/AAAI.
- [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.
- [Martinez-Araiza and López-Mellado, 2016] Ulises Martinez-Araiza and Ernesto López-Mellado. Ctl model repair for inter-organizational business processes modelled as owfn. *IFAC-PapersOnLine*, 49(2):6–11, 2016.
- [Meier *et al.*, 2015] Arne Meier, Michael Thomas, Heribert Vollmer, and Martin Mundhenk. Erratum: The complexity of satisfiability for fragments of CTL and ctl^* . *Int. J. Found. Comput. Sci.*, 26(8):1189, 2015.
- [Ramdani *et al.*, 2019] Mohamed Ramdani, Laid Kahloul, Mohamed Khalgui, and Yousra Hafidi. R-tnces rebuilding: A new method of ctl model update for reconfigurable systems. In *Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering*, pages 159–168. SCITEPRESS-Science and Technology Publications, Lda, 2019.
- [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.
- [Tremblay, 1996] Jean-Paul Tremblay. *Logic and Discrete Mathematics: a Computer Science Perspective*. Upper Saddle River, NJ: Prentice Hall, 1996.
- [Wang *et al.*, 2010] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. Forgetting for knowledge bases in DL-Lite. *Annals of Mathematics and Artificial Intelligence*, 58(1-2):117–151, 2010.
- [Wang *et al.*, 2012] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang. Forgetting in logic programs under strong equivalence. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference*, pages 643–647, Rome, Italy, 2012. AAAI Press.
- [Wang *et al.*, 2013] Yisong Wang, Kewen Wang, and Mingyi Zhang. Forgetting for answer set programs revisited. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 1162–1168, Beijing, China, 2013. IJCAI/AAAI.
- [Wong, 2009] Ka-Shu Wong. *Forgetting in Logic Programs*. PhD thesis, The University of New South Wales, 2009.
- [Woodcock and Morgan, 1990] Jim CP Woodcock and Carroll Morgan. Refinement of state-based concurrent systems. In *International Symposium of VDM Europe*, pages 340–351. Springer, 1990.
- [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.
- [Zhang and Zhou, 2008] Yan Zhang and Yi Zhou. Properties of knowledge forgetting. In Maurice Pagnucco and Michael Thielscher, editors, *Proceedings of the Twelfth International Workshop on Non-Monotonic Reasoning*, pages 68–75, Sydney, Australia, 2008.
- [Zhang and Zhou, 2009] Yan Zhang and Yi Zhou. Knowledge forgetting: Properties and applications. *Artificial Intelligence*, 173(16-17):1525–1537, 2009.
- [Zhang *et al.*, 2005] Yan Zhang, Norman Y. Foo, and Kewen Wang. Solving logic program conflict through strong and weak forgettings. In *Ijcai-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, Uk, July 30-August*, pages 627–634, 2005.
- [Zhao and Schmidt, 2017] Yizheng Zhao and Renate A Schmidt. Role forgetting for $alcoqh(\delta)$ -ontologies using an ackermann-based approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1354–1361. AAAI Press, 2017.