On the Weakest Sufficient Conditions in Propositional μ-calculus

No Author Given

No Institute Given

Abstract. The μ-calculus is one of the most important logics describing specifications of transition systems. It has been extensively explored for formal verification in model checking due to its exceptional balance between expressiveness and algorithmic properties. On the one hand, some information content in a specification might become irrelevant or unnecessary due to various reasons from the perspective of knowledge representation. On the other hand, a weakest precondition of a specification is badly necessary in verification, where a (weakest) precondition is sufficient for a transition system to enjoy a desire property. This paper is to address these scenarios for μ-calculus in a principle way in terms of knowledge *forgetting*. In particular, it proposes a notion of forgetting by a generalized bisimilar equivalence (over a signature) and explores its important properties as a knowledge distilling operator, besides some reasoning complexity results. It then shows that how the weakest sufficient condition and the strongest necessary condition can be established via forgetting. It also discusses knowledge update for μ-calculus in terms of forgetting.

Keywords: Weakest precondition · Forgetting · Knowledge update.

1 Introduction

2 Related work

3 Preliminaries

In this section, we introduce the technical and notational preliminaries, i.e., the syntax and semantics of μ-calculus, closely related to this paper. Moreover, throughout this paper, we denote by $\overline{V}$ the complement of $V \subseteq B$ on a given set B, i.e., $\overline{V} = B - V$.

3.1 The syntax of μ-calculus

Modal μ-calculus is an extension of modal logic, and we consider the propositional μ-calculus introduced by Kozen [8]. Let $\mathcal{A} = \{p, q, \dots\}$ be a set of propositional letters (atoms) and $\mathcal{V} = \{X, Y, \dots\}$ be a set of variables. Then, the formulas of the μ-calculus, called μ-formulas (or formulas), over these sets can be inductively defined in Backus-Naur form:

$$\varphi := p \mid \neg p \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathrm{EX}\varphi \mid \mathrm{AX}\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $p \in \mathcal{A}$ and $X \in \mathcal{V}$. $\top$ and $\bot$ are also μ-calculus formulas, which express 'true' and 'false', respectively.

It is obvious that negations, i.e., '$\neg$', are allowed only before propositional letters. All the results presented here extend to the general case where negation before variables is also allowed, restricted as usual to positive occurrences of bound variables; that is variables appear after an even number of negations. Variables, propositional letters and their negations are called *literals*. For convenience, in the following, φ, $\varphi_1, \ldots, \psi$, ψ_1, $\ldots$ are used to denote μ-formulas. By *Var*(φ) we mean the set of atoms appearing in formula φ.

A formula is *well named* iff every variable is bound at most once in the formula, and free variables are distinct from bound variables. For a variable X bound in a well named formula φ there exists a unique subterm of φ of the form $\delta X.\varphi(X)$ with $\delta \in \{\nu, \mu\}$.

Variable X in $\delta X.\varphi(X)$ is *guarded* iff every occurrence of X in $\varphi(X)$ is within the scope of some modality operators EX or AX. A formula is guarded iff every bound variable in the formula is guarded. Furthermore, a μ-*sentence* is a formula containing no free variables, i.e., no variables unbound by an operator.

In the following, we restrict ourselves to **guarded**, **well-named** μ-sentences.

3.2 The semantics of μ-calculus

Generally, μ-formulas are interpreted in transition systems of the form $\mathcal{M} = (S, r, R, L)$, which we call a Kripke structure, where:

- S is a nonempty set of states,
- $r \in S$,
- R is a binary relation on S, i.e. $R \subseteq S \times S$, called a transition relation, and
- $L : S \to 2^{\mathcal{A}}$ is a labeling function.

Sometimes, r is called the '*root*' of $\mathcal{M}$ [4]. A Kripke structure $\mathcal{M}$ is finite if S is finite and $q \notin L(s)$ (for each state $s \in S$) for almost all $q \in \mathcal{A}$.

Given a Kripke structure $\mathcal{M}$ and a valuation $v : \mathcal{V} \to 2^S$, the set of states in which a formula φ is true, denoted as $\|\varphi\|_v^{\mathcal{M}}$, is defined inductively as follows (the superscript $\mathcal{M}$ is omitted when doing so causes no ambiguity):

$$
\begin{aligned}
&\|p\|_v = \{s \mid p \in L(s)\}\ ;\ \|\top\|_v = S\ ;\ \|\bot\|_v = \emptyset;\\
&\|\neg p\|_v = S - \|p\|_v\ ;\\
&\|X\|_v = v(X);\\
&\|\varphi_1 \vee \varphi_2\|_v = \|\varphi_1\|_v \cup \|\varphi_2\|_v\ ;\\
&\|\varphi_1 \wedge \varphi_2\|_v = \|\varphi_1\|_v \cap \|\varphi_2\|_v\ ;\\
&\|\mathrm{EX}\varphi\|_v = \{s | \exists s'.(s, s') \in R \wedge s' \in \|\varphi\|_v\};\\
&\|\mathrm{AX}\varphi\|_v = \{s | \forall s'.(s, s') \in R \Rightarrow s' \in \|\varphi\|_v\};\\
&\|\mu X.\varphi\|_v = \bigcap\{S' \subseteq S |\ \|\varphi\|_{v[X:=S']} \subseteq S'\};\\
&\|\nu X.\varphi\|_v = \bigcup\{S' \subseteq S | S' \subseteq \|\varphi\|_{v[X:=S']}\}.
\end{aligned}
$$

where $v[X := S']$ is the same as the valuation function v except that S' is assigned to X, i.e., for each $x \in \mathcal{V}$:

$$v[X := S'](x) = \begin{cases} S', & \text{if } x = X; \\ v(x), & \text{otherwise.} \end{cases}$$

In the following, we denote $s \in \|\varphi\|_v$ by $(\mathcal{M}, s, v) \models \varphi$ and we may leave out the valuation v if φ is a μ-sentence. $(\mathcal{M}, v) \models \varphi$ is used to denote $(\mathcal{M}, r, v) \models \varphi$. $(\mathcal{M}, v)$ is a model of φ whenever $(\mathcal{M}, v) \models \varphi$. In this case, *Mod*$(\varphi)$ denotes the set of models of φ. Particularly, if φ is a μ-sentence, then we use $\mathcal{M} \models \varphi$ to replace $(\mathcal{M}, v) \models \varphi$ and *Mod*$(\varphi) = \{\mathcal{M} \mid \mathcal{M} \models \varphi\}$. Similarly, let Σ be a set of μ-sentences; we define *Mod*(Σ) as the set of Kripke structures $\mathcal{M}$ such that $\mathcal{M} \models \varphi$ for each $\varphi \in \Sigma$. Moreover, ψ is a *logical consequence* of φ, denoted by $\varphi \models \psi$, if $(\mathcal{M}, v) \models \varphi$ then $(\mathcal{M}, v) \models \psi$ for every Kripke structure $\mathcal{M}$ and valuation v. Particularly, given two sentences (or set of sentences) Σ and Π, $\Sigma \models \Pi$ if *Mod*$(\Sigma) \subseteq$ *Mod*(Π). And $\Sigma \equiv \Pi$ whenever *Mod*$(\Sigma) =$ *Mod*(Π); in this case we also call Σ and Π *semantically equivalent*.

A formula ϕ is *irrelevant to* the atoms in a set V (or simply V*-irrelevant*), written IR(ϕ, V), if there is a formula ψ with *Var*$(\psi) \cap V = \emptyset$ such that $\phi \equiv \psi$. The V-*irrelevance* of a set of formulas can be defined similarly, i.e., a set Σ of formulas is irrelevant to the atoms in V, written IR(Σ, V), if IR(φ, V) for each $\varphi \in \Sigma$.

3.3 Disjunctive μ-formula

An alternative syntax for the μ-calculus, called *covers-syntax*, is obtained by substituting the EX operator with a set of *cover operators*, one for each natural n. In this way, $Cover(\emptyset)$ is a μ-formula and for $n \geq 1$, if $\varphi_1, \ldots, \varphi_n$ are formulas, then

$$Cover(\varphi_1, \ldots, \varphi_n)$$

is a formula. For a given Kripke structure $\mathcal{M} = (S, r, R, L)$, $Cover(\emptyset)$ is true in $\mathcal{M}$ if and only if the root of $\mathcal{M}$ does not have any successor, while $Cover(\varphi_1, \ldots, \varphi_n)$ is true in $\mathcal{M}$ if and only if the successors of the root are covered by $\varphi_1, \ldots, \varphi_n$. More formally, $(\mathcal{M}, s, v) \models Cover(\varphi_1, \ldots, \varphi_n)$ with $s \in S$ if and only if:

- for every $i = 1, ..., n$, there exists t with $(s, t) \in R$ and $(\mathcal{M}, t, v) \models \varphi_i$;
- for every t with $(s, t) \in R$ there exists $i \in \{1, ..., n\}$ with $(\mathcal{M}, t, v) \models \varphi_i$.

It has shown that the μ-calculus obtained from the covers-syntax is equivalent to the familiar μ-calculus talked in subsection 3.1[5].

The disjunctive formula of μ-formula originate from the work in [7]. In this paper, we use the definition of disjunctive formula in [5].

Definition 1 (disjunctive formula [5]). *The set of disjunctive formulas, $\mathcal{F}_d$ is the smallest set containing $\top$, $\bot$, and non-contradictory conjunciton of literals which is closed under:*

(1) disjunctions;
(2) special conjunctions: if $\varphi_1, \ldots, \varphi_n \in \mathcal{F}_d$ and δ is a non-contradictory conjunction of literals, then $\delta \wedge Cover(\varphi_1, \ldots, \varphi_n) \in \mathcal{F}_d$;

(3) fixpoint operators: if $\varphi \in \mathcal{F}_d$, φ does not contain $X \wedge \psi$ as a subformula for any formula ψ, and X is positive in φ, then $\mu X.\varphi$ and $\nu X.\varphi$ are in $\mathcal{F}_d$.

The disjunctive formulas are representative of the whole μ-calculus, i.e., any μ-calculus formula is equivalent to a disjuntive one.

4 Forgetting in μ-calculus

As has been shown in [6], the WSC of a given CTL formula (property) under a set of atoms and a non-terminating system (called an initial K-structure) can be computed by using the forgetting technique. However, that work does not discuss how to obtain the WSC when the given property is a μ-formula. In this section, we extend the forgetting in CTL to μ-calculus from two aspects: (1) the language discussed is extended from CTL to μ-calculus; and (2) the Kripke structures are more general, i.e., the Kripke structures can contain infinite states, multiple initial states, and so on.

In particular, we present the definition of forgetting in μ-calculus and investigate its semantic properties in this section. First, we give the definition of V-bisimulation between Kripke structures, in which $V \subseteq \mathcal{A}$ is a set of atoms. The notion of V-bisimulation captures the idea that the two systems are behaviourally the same except for the atoms in V. In this way, we define forgetting by using V-bisimulation.

Second, the related properties (i.e., modularity, commutativity, and homogeneity) of the forgetting operator are explored. Finally, it shows that the model checking problem of forgetting V from a disjunctive formula is in NP $\cap$ co-NP, and the reasoning problems are EXPTIME-complete.

4.1 Definition of Forgetting

Recalling the meaning of forgetting in the explored logic languages, "forgetting" some atoms from a given formula should not violate the existing specification over the remaining signature. That is, the models of the formula will be extended to some other Kripke structures such that those Kripke structures simulate the existing models on the remaining signature and vice versa. This reminds us to think about the notion of *bisimulation*.

A bisimulation is a binary relation between state transition systems (they are expressed by Kripke structures in this paper) and associating systems that behave in the same way, in the sense that two systems mimic each other. More clearly, if two Kripke structures $\mathcal{M}_1$ and $\mathcal{M}_2$ are bisimilar, then they satisfy the same formula, i.e., for each formula φ, $\mathcal{M}_1 \models \varphi$ iff $\mathcal{M}_2 \models \varphi$. The result is that neither of the systems can be distinguished from the other by an observer

To clarify the meaning of "forgetting" presented earlier, we define the bisimilar relation on a given signature between Kripke structures. That is, we extend the bisimulation to one under a given set of atoms, i.e., V-bisimulation with V being a set of atoms. For convenience, let $\mathcal{M}_i = (S_i, r_i, R_i, L_i)$ with $i \in \mathbf{N}$ be Kripke structures.

Definition 2 (V-bisimulation). *Let $V \subseteq \mathcal{A}$ and $\mathcal{M}_1$ and $\mathcal{M}_2$ be two Kripke structures. $\mathcal{B} \subseteq S_1 \times S_2$ is a V-bisimulation between $\mathcal{M}_1$ and $\mathcal{M}_2$ if:*

- $r_1 \mathcal{B} r_2$,
- *for each* $s \in S_1$ *and* $t \in S_2$, *if* $s\mathcal{B}t$ *then* $p \in L_1(s)$ *iff* $p \in L_2(t)$ *for each* $p \in \mathcal{A}-V$,
- $(s, s') \in R_1$ *and* $s\mathcal{B}t$ *imply that there is a* t' *such that* $s'\mathcal{B}t'$ *and* $(t, t') \in R_2$, *and*
- *vice versa: if* $s\mathcal{B}t$ *and* $(t, t') \in R_2$, *then there is an* s' *with* $(s, s') \in R_1$ *and* $t'\mathcal{B}s'$.

On the one hand, the V-bisimulation is the same with the $\mathcal{L}$-bisimulation[1] in [3], but on the complement. In this case, as was stated in [3] that any $\mathcal{L}$-sentence φ (that is, a μ-sentence that uses only symbols from the language $\mathcal{L}$) is invariant for $\mathcal{L}$-bisimulation, i.e., if there is an $\mathcal{L}$-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$, then φ holds in $\mathcal{M}$ iff it holds in $\mathcal{M}'$. Therefore, if $\text{IR}(\varphi, V)$ and $\mathcal{M} \leftrightarrow_V \mathcal{M}'$, then φ holds in $\mathcal{M}$ iff it holds in $\mathcal{M}'$. In this paper, we call this property V*-invariant*.

On the other hand, it is easy to see that our definition is similar to that introduced in [6]. That is, the definitions are same whenever $(\mathcal{M}_i, r_i)$ is limited to an initial K-structure. Moreover, the V-bisimulation defined in [6], the classical bisimulation-equivalence of Definition 7.1 in [1], the state equivalence (i.e., E_n) in [2], and the state-based bisimulation notion of Definition 7.7 in [1] are closely related.[2] In this sense, one can see that our V-bisimulation is also closely related to those definitions to some extent.

Two Kripke structures $\mathcal{M}_1$ and $\mathcal{M}_2$ are V*-bisimilar*, denoted as $\mathcal{M}_1 \leftrightarrow_V \mathcal{M}_2$, if there exists a V-bisimulation $\mathcal{B}$ between them. In this case, $\mathcal{M}_1$ and $\mathcal{M}_2$ are bisimilar on V. To obtain some intuition of the V-bisimulation, let us consider the following example.

Example 1. In Fig.1, we can check that $\mathcal{M} \leftrightarrow_{\{ch\}} \mathcal{M}'$ because there is a $\{ch\}$-bisimulation $\mathcal{B} = \{(s_0, t_0), (s_1, t_1), (s_2, t_1)\}$ between $\mathcal{M}$ and $\mathcal{M}'$.

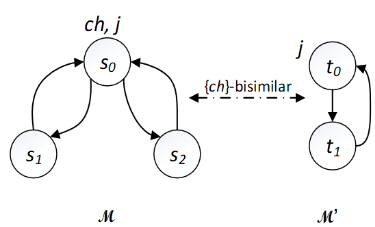

Fig.1. Two $\{ch\}$-bisimilar Kripke structures.

Moreover, we can see that the relation $\leftrightarrow_V$ has some interesting properties in addition to the equivalence relation. Formally:

Proposition 1. *Let* $V, V_1 \subseteq \mathcal{A}$ *and* $\mathcal{M}_1$, $\mathcal{M}_2$ *and* $\mathcal{M}_3$ *be three Kripke structures, then we have:*

(i) $\leftrightarrow_V$ *is an equivalence relation between Kripke structures;*

[1] It is a relation satisfying the clauses in Definition 2 just for the symbols in language $\mathcal{L}$

[2] The V-bisimulation defined in [6] is similar to the state equivalence (i.e., E_n) in [2], yet it is different in the sense that the one in [6] is defined on K-structures, while state equivalence is defined on states. Moreover, V-bisimulation is different from the state-based bisimulation notion of Definition 7.7 in [1], which is defined for states of a given K-structure.

(ii) if $\mathcal{M}_1 \leftrightarrow_V \mathcal{M}_2$ *and* $\mathcal{M}_2 \leftrightarrow_{V_1} \mathcal{M}_3$*, then* $\mathcal{M}_1 \leftrightarrow_{V \cup V_1} \mathcal{M}_3$.

Intuitively, property (i) in Proposition 1 means that $\leftrightarrow_V$ is reflexive, symmetric, and transitive. (ii) indicates that if a Kripke structure is V and V_1-bisimilar to the other two Kripke structures respectively, then those two Kripke structures are $V \cup V_1$-bisimilar. As we will show in the following context, it is important to demonstrate the *modularity*, one of the important properties of forgetting in μ-calculus.

We now define forgetting in μ-calculus.

Definition 3 (Forgetting). *Let* $V \subseteq \mathcal{A}$ *and* ϕ *be a* μ*-sentence. A* μ*-sentence* ψ *with* $Var(\psi) \cap V = \emptyset$ *is a* result of forgetting V from ϕ *if*

$$Mod(\psi) = \{\mathcal{M} \mid \exists \mathcal{M}' \in Mod(\phi) \;\&\; \mathcal{M}' \leftrightarrow_V \mathcal{M}\}.$$

We denote the result of forgetting V from ϕ as $\mathrm{F}_\mu(\phi, V)$. It is not difficult to see that Definition 3 implies that if both ψ and ψ' are results of forgetting V from ϕ, then $Mod(\psi) = Mod(\psi')$, i.e., ψ and ψ' have the same models. In this sense, the result of forgetting V from ϕ is unique up to semantic equivalence.

It is worthy of note that D'Agostino at al. studied the notion of *uniform interpolation* in μ-calculus, and inicated that μ-calculus has the uniform interpolation property [3,5]. Informally, this means that for every μ-sentence φ and every finite set $V \subseteq Var(\varphi)$, there exists a μ-sentenc $\widetilde{\exists} V \varphi$ which does not contain atoms from V but is logically closest to φ in some sense.

We should mention that our forgetting definition $\mathrm{F}_\mu(\phi, V)$ is equivalent to the semantic definition of formula in [5].

References

1. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
2. Browne, M.C., Clarke, E.M., Grümberg, O.: Characterizing finite kripke structures in propositional temporal logic. Theoretical Computer Science **59**(1-2), 115–131 (1988)
3. D'Agostino, G., Hollenberg, M.: Uniform interpolation, automata and the modal μ-calculus. In: Kracht, M., de Rijke, M., Wansing, H., Zakharyaschev, M. (eds.) Advances in Modal Logic 1, papers from the first workshop on "Advances in Modal logic," held in Berlin, Germany, 8-10 October 1996. pp. 73–84. CSLI Publications (1996)
4. D'Agostino, G., Hollenberg, M.: Logical questions concerning the μ-calculus: Interpolation, lyndon and los-tarski. J. Symb. Log. **65**(1), 310–332 (2000). https://doi.org/10.2307/2586539, https://doi.org/10.2307/2586539
5. D'Agostino, G., Lenzi, G.: On modal μ-calculus with explicit interpolants. Journal of applied logic **4**(3), 256–278 (2006)
6. Feng, R., Acar, E., Schlobach, S., Wang, Y., Liu, W.: On sufficient and necessary conditions in bounded CTL: A forgetting approach. In: Calvanese, D., Erdem, E., Thielscher, M. (eds.) Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020. pp. 361–370 (2020). https://doi.org/10.24963/kr.2020/37, https://doi.org/10.24963/kr.2020/37
7. Janin, D., Walukiewicz, I.: Automata for the modal μ-calculus and related results. In: International Symposium on Mathematical Foundations of Computer Science. pp. 552–562. Springer (1995)
8. Kozen, D.: Results on the propositional μ-calculus. Theoretical Computer Science **27**, 333–354 (1983)

A Supplementary Material: Proof Appendix