National Institute of
Advanced Industrial Science
and Technology
AIST

AIST-PS-2008-009

Pre- and Post-conditions Expressed in Variants of the Modal μ-calculus

Yoshinori Tanabe Toshifusa Sekizawa
Yoshifumi Yuasa Koichi Takahashi

Research Center for Verification and Semantics (CVS)
National Institute of Advanced Industrial Science and Technology (AIST)

算譜科学研究速報
Programming Science
Technical Report

Research Center
for Verification and Semantics

独立行政法人 産業技術総合研究所 システム検証研究センター 発行
Published by Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology

Pre- and Post-conditions Expressed in Variants of the Modal μ-calculus

Yoshinori Tanabe Toshifusa Sekizawa
Yoshifumi Yuasa Koichi Takahashi

Research Center for Verification and Semantics (CVS)
National Institute of Advanced Industrial Science and Technology (AIST)

Abstract

Properties of Kripke structures can be expressed by formulas of the modal μ-calculus. Despite its strong expressive power, the validity problem of the modal μ-calculus is decidable, and so are some of its variants enriched by inverse programs, graded modalities, and nominals. In this paper, we show that pre- and post-conditions of transformations of Kripke structures, such as addition/deletion of states and edges, can be expressed using variants of the modal μ-calculus, some of which are validity-decidable. As an application, we utilize them to verify the properties of pointer manipulating programs.

1 Introduction

In previous studies, we applied temporal logics to verification problems in some areas, such as concurrent garbage collection [1] and one-dimensional cellular automata [2]. The target of the studies are considered as graph transformation systems, and the basic idea of the analysis is to regard the graphs as Kripke structures and express their properties using formulas of temporal logics such as computational tree logic (CTL). They have expressive power to describe the properties of the systems and their validity problems are decidable. Although CTL has been successfully applied to the above-mentioned target systems, we need more expressive power to undertake similar approaches for more complicated systems.

First, we use general fixed-point operators, which play a key role in expressing graph properties such as reachability. While CTL has fixed-point operators, which is the main reason why we employed this logic as the tool for analysis, its usage is restricted to some fixed patterns such as EU or AG. Using general fixed-point operators μ and ν, one can express more complicated properties.

Second, we use *nominals* [3], which are a type of atomic formulas but are satisfied by one and only one node in a Kripke structure. Nominals can be used, for example, to express pointer-type variables of a programming language — when a state of a Kripke structure satisfies a nominal, the state is regarded as the value of the corresponding variable. A propositional symbol cannot be substituted for a nominal since it may be satisfied by two or more states while the value of a variable should be unique.

The third point is regarding functional modalities. While an ordinary modality m is interpreted in a Kripke structure as a relation $R(m)$, a functional modality f is interpreted as a (partial) function $R(f)$; that is, for each state s, there is at most one s' such that $(s, s') \in R(f)$. They can be used to express, for example, pointer-type fields of a structure in a C-like language just as nominals express pointer-type variables.

The fourth point is *backward modalities*.[*1] A backward modality m^{-1}, where m is an ordinary (forward) modality, follows the transition relation of a Kripke structure in the reverse direction. We have already used them in [2], and they are vital for our computation of the weakest precondition as we will see in Section 3.

[*1] While the term *inverse programs* might be more commonly used (e.g., [4]), it may cause confusion in this paper since we use the word *program* with a different meaning.

Thus, our logic $\mathcal{L}$ has nominals and functional and backward modalities. It can be considered as a variant of *enriched μ-calculi* [4]. Formulas of the logic express properties of Kripke structures.

Pre- and post-conditions play an important role when we reason the properties of Kripke structures with regard to programs that transform them. We list basic transformations of Kripke structures, such as addition of states or modification of transition relations, and show that the weakest preconditions can also be expressed in $\mathcal{L}$. Although $\mathcal{L}$ is not validity-decidable [5], there is a sound (but naturally incomplete) decision procedure for validity. Thus, backward reasoning is possible. This part is an extension of our previous work [6].

We also consider a sublogic $\mathcal{L}'$ of $\mathcal{L}$ by removing the backward modalities. Logic $\mathcal{L}'$ is validity-decidable [4]. We show that the strongest postcondition of each statement can be defined in $\mathcal{L}$, and if a property is expressed in $\mathcal{L}'$, so is its strongest postcondition. In this case, the complete decision procedure [4] can also be used.

Using forward reasoning, we illustrate how the properties of pointer manipulating programs are verified. Various studies have analyzed programs manipulating pointers. In one of the approaches, a three-valued logic is used in addition to the first-order logic enhanced with an operator to take the transitive closure [7]. Another approach uses Separation Logic [8], which is an extension of the Hoare logic and has operators to handle the status of the heap. Our approach differs from them in that we use an existing logic with a decision procedure for validity testing.

The rest of the paper is organized as follows. In Section 2 we define the syntax and semantics of the logic, and introduce transformations of Kripke structures. Preconditions, which are used in backward reasoning, are introduced in Section 3, while postconditions for forward reasoning are discussed in Section 4. We show an verification example in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Syntax

Let PS, Nom, PV, GMS, FMS be countable sets of propositional symbols, nominals, propositional variables, general modality symbols, and functional modality symbols, respectively. Modalities and formulas are defined as follows.

Mod $\ni m ::= \mathbf{o} \mid g \mid f \mid g^{-1} \mid f^{-1}$

Form $\ni \varphi ::= \mathbf{false} \mid p \mid x \mid X \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle m \rangle \varphi \mid \mu X \varphi$,

where $p \in$ PS, $x \in$ Nom, $X \in$ PV, $g \in$ GMS, and $f \in$ FMS. In $\mu X \varphi$, all free occurrences of X in φ must be positive. The symbol $\mathbf{o}$ is called the *global modality.* A modality in the form of m^{-1} is called a *backward modality.* We denote by $\mathcal{L}$ the set of formulas and by $\mathcal{L}'$ the set of formulas that do not contain backward modalities. We define Atom = PS $\cup$ Nom and MS = GMS $\cup$ FMS. In the rest of the paper we assume that Nom contains an element called **nil**.

The following standard abbreviations are used: $\mathbf{true} = \neg\mathbf{false}$, $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$, $[m]\varphi = \neg\langle m \rangle \neg\varphi$, $\nu X \varphi = \neg\mu\neg\varphi[\neg X/X]$.

2.2 Semantics

A *Kripke structure* for $\mathcal{L}$ or $\mathcal{L}'$ is a tuple $\mathcal{K} = (S, R, L, \mathrm{nil})$ that satisfies the following conditions. We denote by $\mathscr{P}(S)$ the powerset of S.

- nil is an element of set S.
- $R : \mathrm{MS} \rightarrow \mathscr{P}(S \times S)$. $R(f)$ is a (partial) function if $f \in$ FMS.
- $L : \mathrm{Atom} \rightarrow \mathscr{P}(S)$. $L(x)$ is a singleton if $x \in$ Nom.
- $L(\mathbf{nil}) = \{\mathrm{nil}\}$; nil $\notin L(p)$ for $p \in$ PS; and $(\mathrm{nil}, s) \notin R(m)$ and $(s, \mathrm{nil}) \notin R(m)$ for $s \in S$ and $m \in$ MS.

For $x \in$ Nom, we denote by $L'(x)$ the unique element of $L(x)$, that is, $L(x) = \{L'(x)\}$. For $f \in$ FMS and $s \in S$, we define $R(f, s) \in S$ by $R(f, s) = s'$ if there exists $s' \in S$ such that $(s, s') \in R(f)$; and

Table. 1 Transformations of Kripke structures

τ	Condition for $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$
$\mathrm{mvNom1}(x_1, x_2)$	$L_2 = L_1[x_1 \mapsto L_1(x_2)]$
$\mathrm{mvNom2}(x_1, x_2, f)$	$L_2 = L_1[x_1 \mapsto R_1(f, L_1(x_2))]$
$\mathrm{addLab}(x, p)$	$L_2 = L_1[p \mapsto L_1(p) \cup L_1(x)]$
$\mathrm{delLab}(x, p)$	$L_2 = L_1[p \mapsto L_1(p) \setminus L_1(x)]$
$\mathrm{addTrans}(m, x_1, x_2)$	$R_2 = R_1[m \mapsto R_1(m) \cup \{(L_1(x_1), L_1(x_2))\}]$
$\mathrm{delTrans}(m, x_1, x_2)$	$R_2 = R_1[m \mapsto R_1(m) \setminus \{(L_1(x_1), L_1(x_2))\}]$
$\mathrm{delTrans}(f, x)$	$R_2 = R_1[f \mapsto R_1(f) \setminus \{(L_1(x), R_1(f, L_1(x)))\}]$
$\mathrm{addState}(x)$	$\exists s \in S_2.\, S_2 = S_1 \uplus \{s\},\ L_2 = L_1[x \mapsto \{s\}]$
$\mathrm{delState}(x)$	$S_2 = S_1 \setminus \{L_1(x)\},\ R_2 = R_1 \setminus ((\{L_1(x)\} \times S_1) \cup (S_1 \times \{L_1(x)\})),$ $L_2 = L_1[y \mapsto \{\mathrm{nil}_1\} \mid L_1(y) = L_1(x)]$

$R(f, s) = \mathrm{nil}$ otherwise.

We extend R so that $R(m)$ is defined for all $m \in \mathrm{Mod}$:

- $R(\mathbf{o}) = S \times S$
- $R(m^{-1}) = (R(m))^{-1}$ for $m \in \mathrm{MS}$.

A function $\rho : \mathrm{PV} \to \mathscr{P}(S)$ is called a *valuation* for $\mathcal{K}$. The *interpretation* $[\![\varphi]\!]^{\mathcal{K},\rho} \subseteq S$ of $\varphi \in \mathcal{L}$ is defined in the standard way as follows. Symbols $\mathcal{K}$ and/or ρ can be omitted if no confusion occurs. For function F, we denote by $F[a \to b]$ a function G defined by $\mathrm{dom}(G) = \mathrm{dom}(F) \cup \{a\}$, $G(a) = b$, and $G(x) = F(x)$ for $x \in \mathrm{dom}(F) \setminus \{a\}$.

- $[\![\mathbf{false}]\!] = \varnothing$.
- $[\![a]\!] = L(a)$ for $a \in \mathrm{Atom}$.
- $[\![X]\!] = \rho(X)$ for $X \in \mathrm{PV}$.
- $[\![\neg\varphi]\!] = S \setminus [\![\varphi]\!]$.
- $[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$.
- $[\![\langle m \rangle \varphi]\!] = \{s \in S \mid \exists s' \in S.\, (s, s') \in R(m) \text{ and } s' \in [\![\varphi]\!]\}$
- $[\![\mu X \varphi]\!] = \bigcap\{T \subseteq S \mid [\![\varphi]\!]^{\rho[X \mapsto T]} \subseteq T\}$

We write $\mathcal{K}, \rho, s \models \varphi$ if $s \in [\![\varphi]\!]^{\mathcal{K},\rho}$. Again, $\mathcal{K}$ and/or ρ are often omitted. We write $\mathcal{K} \models \varphi$ if $\mathcal{K}, \rho, s \models \varphi$ holds for any valuation ρ and $s \in S$. Formulas φ and φ' are *equivalent* ($\varphi \equiv \varphi'$) if $[\![\varphi]\!]^{\mathcal{K},\rho} = [\![\varphi']\!]^{\mathcal{K},\rho}$ for any Kripke structure $\mathcal{K}$ and valuation ρ. A formula φ is *valid* if it is equivalent to **true**.

For nominal x and formulas φ, φ_1, and φ_2, we define $@x\,\varphi = \langle \mathbf{o} \rangle (x \wedge \varphi)$ and $\varphi_1 \to \varphi_2\,; \varphi_3 = (\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \varphi_3)$. Obviously $@x\,\varphi \equiv [\mathbf{o}](x \to \varphi)$ and $\varphi_1 \to \varphi_2\,; \varphi_3 \equiv (\varphi_1 \to \varphi_2) \wedge (\neg\varphi_1 \to \varphi_3)$ holds.

A formula $\xi \in \mathcal{L}$ is *FG-free* if for all its subformulas in the form of $\mu X \psi$, the global modality does not appear in ψ. It is *GV-free* if for all its subformulas in the form of $\langle \mathbf{o} \rangle \psi$, no free variable occurs in ψ. In other words, a formula is FG-free if no *f*ixed-point operator has the *g*lobal modality in its scope, and it is GV-free if no operator with the *g*lobal modality has a free *v*ariable in its scope. Clearly, if a closed formula is FG-free, it is GV-free.

Lemma 1 For any formula $\varphi \in \mathcal{L}$, there is a FG-free formula $\psi \in \mathcal{L}$ that is equivalent to φ.

For a proof, please refer to the appendix.

2.3 Transformations of Kripke Structures

In this section, we introduce several transformations of Kripke structures. Formally, a transformation is defined as a relation on the class of all Kripke structures.

We consider the following transformations of Kripke structures. In the following description, we assume $x, x_1, x_2, y \in \mathrm{Nom}$, $m \in \mathrm{MS}$, $f \in \mathrm{FMS}$, $p \in \mathrm{PS}$, and $\mathcal{K} = (K, R, L, \mathrm{nil})$ is a Kripke structure.

- mvNom1(x_1, x_2): Changes the interpretation of x_1 to $L'(x_2)$.
- mvNom2(x_1, x_2, f): Changes the interpretation of x_1 to $R(f, L'(x_2))$.
- addLab(p, x): Makes $L'(x)$ satisfy p.
- delLab(p, x): Makes $L'(x)$ not satisfy p.
- addTrans(m, x_1, x_2): Adds a transition for m from $L'(x_1)$ to $L'(x_2)$.
- delTrans(m, x_1, x_2): Removes a transition for m from $L'(x_1)$ to $L'(x_2)$ if it exists.
- delTrans(f, x_1): Removes a transition for f from $L'(x_1)$ if it exists.
- addState(x): Adds a state and makes it $L'(x)$.
- delState(x): Removes the state $L'(x)$. Any transition to and from the state is also removed. The interpretation $L'(y)$ of a nominal y becomes nil if the $L'(y)$ was the removed state.

Precise definitions of the transformations are given in Table 1. We assume that $\mathcal{K}_i = (S_i, R_i, L_i, \mathrm{nil}_i)$ $(i = 1, 2)$ are Kripke structures. For each transformation τ, the condition for $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$ is described in the table. Members of the tuple not explicitly referred in the table should be identical. For example, there are implicit conditions for mvNom1(x_1, x_2): $S_1 = S_2$, $R_1 = R_2$, and $\mathrm{nil}_1 = \mathrm{nil}_2$.

3 Preconditions

In this section, for each transformation τ defined in the previous section and formula $\psi \in \mathcal{L}$, we define a formula $\mathrm{wp}(\tau, \psi)$ and show that it is the weakest precondition of ψ with respect to τ.

We begin by introducing a notation $T(\psi) = \mathrm{RD}\{\mathit{desc}\}(\psi)$ to define a formula $T(\psi)$ from given formula ψ by induction on the construction of ψ. We put a list separated by semicolons for *desc*: they correspond to only those cases other than the following: $T(a) = a$ $(a \in \{\mathbf{false}\} \cup \mathrm{Atom} \cup \mathrm{PV})$, $T(\neg\varphi) = \neg T(\varphi)$, $T(\varphi_1 \vee \varphi_2) = T(\varphi_1) \vee T(\varphi_2)$, $T(\langle m \rangle \varphi) = \langle m \rangle T(\varphi)$, and $T(\mu X \varphi) = \mu X\, T(\varphi)$. For example, let $x \in \mathrm{Nom}$. If $T(\psi)$ is defined by $T(\psi) = \mathrm{RD}\{x \rightarrowtail \neg x\ ;\ \langle f \rangle \varphi \rightarrowtail \langle f \rangle (x \wedge \overline{\varphi})\ \ (f \in \mathrm{FMS})\}(\psi)$, it means $T(x) = \neg x$, (but for $x' \in \mathrm{Nom} \setminus \{x\}$, $T(x') = x'$) and $T(\langle f \rangle \varphi) = \langle f \rangle (x \wedge T(\varphi))$ for $f \in \mathrm{FMS}$ (but $T(\langle m \rangle \varphi) = \langle m \rangle T(\varphi)$ for $m \in \mathrm{MS} \setminus \mathrm{FMS}$). Note that symbol $\overline{\varphi}$ is used in *desc* to express $T(\varphi)$. Cases that are not explicitly stated default to the above-mentioned definition; for example $T(\mu X \varphi) = \mu X\, T(\varphi)$.

Using this notation, we define two auxiliary formulas, $\mathrm{wpl}(\tau, \psi)$ and $\mathrm{ns}(x, \psi)$, where $x \in \mathrm{Nom}$. Intuitively, formula $\mathrm{wpl}(\tau, \psi)$ claims that the state satisfies ψ in the Kripke structure transformed by τ; and formula $\mathrm{ns}(x, \psi)$ claims that the newly added state by the transformation addState(x) satisfies the formula ψ. They are defined by induction on the construction of ψ, and $\mathrm{wpl}(\mathrm{addState}(x), \psi)$ and $\mathrm{ns}(x, \psi)$ are defined concurrently. Table 2 defines *desc* for $\mathrm{wpl}(\tau, \psi) = \mathrm{RD}\{\mathit{desc}\}(\psi)$ and $\mathrm{ns}(\tau, \psi) = \mathrm{RD}\{\mathit{desc}\}(\psi)$.

The following lemma is a formal statement for the intuition of wpl and ns described above. Assume φ is a closed formula in $\mathcal{L}$, $\mathcal{K}_i = (S_i, R_i, L_i, \mathrm{nil}_i)$ $(i = 1, 2)$ are Kripke structures, τ is a transformation, and $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$.

Lemma 2 (1) Assume τ is other than addState(x). For any $s \in S_1 \cap S_2$, the following holds.

$$\mathcal{K}_1, s \models \mathrm{wpl}(\tau, \varphi) \iff \mathcal{K}_2, s \models \varphi$$

(2) Assume $x \in \mathrm{Nom}$, $\tau = \mathrm{addState}(x)$, and φ is GV-free. Let $\hat{s} = L_2(x)$, that is, $S_2 = S_1 \uplus \{\hat{s}\}$. Then the following hold.

- $\mathcal{K}_1, s \models \mathrm{wpl}(\tau, \varphi) \iff \mathcal{K}_2, s \models \varphi$ for any $s \in S_1$.
- $\mathcal{K}_2, \hat{s} \models \varphi \iff [\![\mathrm{ns}(x, \varphi)]\!]^{\mathcal{K}_1} = S_1$.
- $\mathcal{K}_2, \hat{s} \not\models \varphi \iff [\![\mathrm{ns}(x, \varphi)]\!]^{\mathcal{K}_1} = \varnothing$.

Please refer to the appendix for a proof.

Using wpl and ns, we define formula wp:

$$\mathrm{wp}(\tau, \varphi) = \begin{cases} \mathrm{wpl}(\tau, \hat{\varphi}) \wedge \mathrm{ns}(x, \hat{\varphi}) & \text{if } \tau = \mathrm{addState}(x) \\ [\mathbf{o}](\neg x \to \mathrm{wpl}(\tau, \varphi)) & \text{if } \tau = \mathrm{delState}(x) \\ \mathrm{wpl}(\tau, \varphi) & \text{otherwise} \end{cases}$$

Table. 2 Definitions of $\mathrm{wpl}(\tau, \psi)$ and $\mathrm{ns}(x, \psi)$

τ	*desc* for $\mathrm{wpl}(\tau, \psi)$
$\mathrm{mvNom1}(x_1, x_2)$	$x_1 \rightarrowtail x_2$
$\mathrm{mvNom2}(x_1, x_2, f)$	$x_1 \rightarrowtail (@x_2\, \langle f \rangle \mathbf{true} \to \langle f^{-1} \rangle x_1\ ; \mathbf{nil})$
$\mathrm{addLab}(x, p)$	$p \rightarrowtail p \vee x$
$\mathrm{delLab}(x, p)$	$p \rightarrowtail p \wedge \neg x$
$\mathrm{addTrans}(m, x_1, x_2)$	$\langle m \rangle \varphi \rightarrowtail \langle m \rangle \overline{\varphi} \vee (x_1 \wedge @x_2\, \overline{\varphi})$; $\langle m^{-1} \rangle \varphi \rightarrowtail \langle m^{-1} \rangle \overline{\varphi} \vee (x_2 \wedge @x_1\, \overline{\varphi})$
$\mathrm{delTrans}(m, x_1, x_2)$	$\langle m \rangle \varphi \rightarrowtail (\neg x_1 \wedge \langle m \rangle \overline{\varphi}) \vee (\langle m \rangle (\neg x_2 \wedge \overline{\varphi}))$; $\langle m^{-1} \rangle \varphi \rightarrowtail (\neg x_2 \wedge \langle m^{-1} \rangle \overline{\varphi}) \vee (\langle m^{-1} \rangle (\neg x_1 \wedge \overline{\varphi}))$
$\mathrm{delTrans}(f, x)$	$\langle f \rangle \varphi \rightarrowtail \neg x \wedge \langle f \rangle \overline{\varphi}$; $\langle f^{-1} \rangle \varphi \rightarrowtail \langle f^{-1} \rangle (\neg x \wedge \overline{\varphi})$
$\mathrm{addState}(x)$	$x \rightarrowtail \mathbf{false}$; $\langle \mathbf{o} \rangle \varphi \rightarrowtail \mathrm{ns}(x, \varphi) \vee \langle \mathbf{o} \rangle \overline{\varphi}$
$\mathrm{delState}(x)$	$x' \rightarrowtail (@x\, x' \to \mathbf{nil}\ ; x')\ \ (x' \in \mathrm{Nom})$; $\langle m \rangle \varphi \rightarrowtail \langle m \rangle (\overline{\varphi} \wedge \neg x)\ \ (m \in \mathrm{MS})$

desc for $\mathrm{ns}(x, \psi)$
$x \rightarrowtail \mathbf{true}$; $a \rightarrowtail \mathbf{false}\ \ (a \in \mathrm{Atom} \setminus \{x\})$; $\langle \mathbf{o} \rangle \varphi \rightarrowtail \overline{\varphi} \vee \langle \mathbf{o} \rangle \mathrm{wpl}(\mathrm{addState}(x), \varphi)$; $\langle m \rangle \varphi \rightarrowtail \mathbf{false}\ \ (m \in \mathrm{Mod} \setminus \{\mathbf{o}\})$

where $\hat{\varphi}$ is a FG-free formula that is equivalent to φ. Its existence is guaranteed by Lemma 1.

Formula $\mathrm{wp}(\tau, \varphi)$ can be regarded as the weakest precondition of φ with respect to τ:

Theorem 3

$$\mathcal{K}_1 \models \mathrm{wp}(\tau, \varphi) \iff \mathcal{K}_2 \models \varphi$$

Proof. We only show in the case $\tau = \mathrm{addState}(x)$, others can be shown similarly.

Since $\hat{\varphi}$ is a closed FG-free formula, it is GV-free. Therefore Lemma 2 can be applied. Assume $\mathcal{K}_1 \models \mathrm{wp}(\tau, \varphi)$ and $s \in S_2$. If $s \in S_1$, since $\mathcal{K}_1, s \models \mathrm{wpl}(\tau, \varphi)$, we have $\mathcal{K}_2, s \models \varphi$. If $s \notin S_1$, that is, if $s = \hat{s}$, $\mathcal{K}_2, s \models \varphi$ also holds. Thus we have $\mathcal{K}_2 \models \varphi$. The other direction is similar. ■

Thus, we can calculate the weakest precondition within the logic $\mathcal{L}$. Although $\mathcal{L}$ is not validity-decidable [5], sound (but incomplete) decision procedures can be built. Combined with such procedures, we can reason the properties of Kripke structures with respect to transformations.

We have defined a sublogic $\mathcal{L}'$ of $\mathcal{L}$. It is desirable to find a formula in $\mathcal{L}'$ with the property of Theorem 3 because $\mathcal{L}'$ is validity-decidable [4]. Our current wp() does not always produce formulas in $\mathcal{L}'$ since it uses backward modalities for mvNom2. A question arises whether it is possible to find an equivalent formula within $\mathcal{L}'$. Unfortunately, the answer is negative. To see this, let us recall some definitions.

Relation $H \subseteq S_1 \times S_2$ is a *simulation* for $\mathcal{L}'$ from $\mathcal{K}_1$ to $\mathcal{K}_2$ if (1) for any $s_1, s_1' \in S_1$, $s_2 \in S_2$, $m \in \mathrm{MS}$ such that $(s_1, s_1') \in R_1(m)$ and $(s_1, s_2) \in H$, there exists $s_2' \in S_2$ such that $(s_2, s_2') \in R_2(m)$ and $(s_1', s_2') \in H$, and (2) for any $s_1 \in S_1$, $s_2 \in S_2$ and $a \in \mathrm{Atom}$ such that $(s_1, s_2) \in H$, $s_1 \in L_1(a) \iff s_2 \in L_2(a)$ holds. Relation H is a *bisimulation* for $\mathcal{L}'$ between $\mathcal{K}_1$ and $\mathcal{K}_2$ if H and H^{-1} is a simulation. The following lemma is well-known.

Lemma 4 Assume $\mathcal{K}_1$ and $\mathcal{K}_2$ are Kripke structures, $s_1 \in S_1$, $s_2 \in S_2$, φ is a closed formula in $\mathcal{L}'$, and H is a bisimulation for $\mathcal{L}'$ between $\mathcal{K}_1$ and $\mathcal{K}_2$. If $(s_1, s_2) \in H$, we have $\mathcal{K}_1, s_1 \models \varphi \iff \mathcal{K}_2, s_2 \models \varphi$.

Let $x, y, z \in \mathrm{Nom}$, $f \in \mathrm{FMS}$, $\varphi_1 = @x\, \langle f \rangle z$, $\tau = \mathrm{mvNom2}(z, y, f)$, and $\varphi_2 = \mathrm{wpl}(\tau, \varphi_1) = @x\, \langle f \rangle \langle f^{-1} \rangle y$.

Proposition 5 There is no formula in $\mathcal{L}'$ that is equivalent to φ_2.

Proof. We define two Kripke structures $\mathcal{K}_i = (S_i, R_i, L_i, \mathrm{nil}_i)$ $(i = 1, 2)$ by $S_1 = \{s_x, s_y, s_1, \mathrm{nil}_1\}$, $R_1(f) = \{(s_x, s_1), (s_y, s_1)\}$, $L_1(x) = \{s_x\}$, $L_1(y) = \{s_y\}$, $S_2 = \{t_x, t_y, t_1, t_2, \mathrm{nil}_2\}$, $R_2(f) = \{(t_x, t_1), (t_y, t_2)\}$, $L_2(x) = \{t_x\}$, and $L_2(y) = \{t_y\}$. Clearly $\mathcal{K}_1 \models \varphi_2$ and $\mathcal{K}_2 \not\models \varphi_2$. However, $H = \{(s_x, t_x), (s_y, t_y), (s_1, t_1), (s_1, t_2), (\mathrm{nil}_1, \mathrm{nil}_2)\}$ is a bisimulation between $\mathcal{K}_1$ and $\mathcal{K}_2$. ■

4 Postconditions

In this section, we discuss postconditions of transformations. We will define formula $\mathrm{post}(\tau, \psi)$ that satisfies the following conditions.

(1) $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$, $\mathcal{K}_1 \models \psi \implies \mathcal{K}_2 \models \mathrm{post}(\tau, \psi)$.
(2) If $\mathcal{K}_2 \models \mathrm{post}(\tau, \psi)$, there exists $\mathcal{K}_1$ such that $\mathcal{K}_1 \models \psi$ and $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$.

It is easy to see that $\mathrm{post}(\tau, \psi)$ is the strongest postcondition of ψ with respect to τ if it satisfies the conditions (1) and (2).

We define two auxiliary formulas $\mathrm{post}_1(\tau)$ and $\mathrm{post}_2(\tau, \psi)$, then post() is defined by $\mathrm{post}(\tau, \psi) = \mathrm{post}_1(\tau) \wedge \mathrm{post}_2(\tau, \psi)$. Formula $\mathrm{post}_1(\tau)$ is defined as follows.[*2] Formula $\mathrm{post}_1(\tau)$ describes obvious properties of the resulting Kripke structures. It is straightforward to see that $\mathrm{post}_1(\tau)$ satisfies the condition (1).

- $\mathrm{post}_1(\mathrm{mvNom1}(x_1, x_2)) = @x_2\, x_1$
- $\mathrm{post}_1(\mathrm{mvNom2}(x_1, x_2, f)) = @x_2\, \langle f \rangle x_1$
- $\mathrm{post}_1(\mathrm{addLab}(x, p)) = @x\, p$
- $\mathrm{post}_1(\mathrm{delLab}(x, p)) = @x\, \neg p$
- $\mathrm{post}_1(\mathrm{addTrans}(m, x_1, x_2)) = @x_1\, \langle m \rangle x_2$
- $\mathrm{post}_1(\mathrm{delTrans}(m, x_1, x_2)) = @x_1\, [m] \neg x_2$
- $\mathrm{post}_1(\mathrm{delTrans}(f, x)) = @x\, [f]\mathbf{false}$
- $\mathrm{post}_1(\mathrm{addState}(x)) =$
$$@x \left((\textstyle\bigwedge_{a \in \mathrm{Atom} \setminus \{x\}} \neg a) \wedge \bigwedge_{m \in \mathrm{MS}} [m]\mathbf{false} \right) \wedge \textstyle\bigwedge_{m \in \mathrm{MS}} [\mathbf{o}][m] \neg x$$
- $\mathrm{post}_1(\mathrm{delState}(x)) = @x\, \mathbf{nil}$

Formula $\mathrm{post}_2(\tau, \psi)$ is defined as follows. Roughly speaking, $\mathrm{post}_2(\tau, \psi)$ is the weakest precondition of ψ with respect to the "reverse transformation" of τ. Therefore, intuitively, conditions (1) and (2) for $\mathrm{post}_2(\tau, \psi)$ should be satisfied.

- $\mathrm{post}_2(\mathrm{mvNom1}(x_1, x_2), \psi) = \mathrm{wpl}(\mathrm{mvNom1}(x_1, y), \psi)$
- $\mathrm{post}_2(\mathrm{mvNom2}(x_1, x_2, f), \psi) = \mathrm{wpl}(\mathrm{mvNom1}(x_1, y), \psi)$
- $\mathrm{post}_2(\mathrm{addLab}(x, p), \psi) = \psi \vee \mathrm{wpl}(\mathrm{delLab}(x, p), \psi)$
- $\mathrm{post}_2(\mathrm{delLab}(x, p), \psi) = \psi \vee \mathrm{wpl}(\mathrm{addLab}(x, p), \psi)$
- $\mathrm{post}_2(\mathrm{addTrans}(m, x_1, x_2), \psi) = \psi \vee \mathrm{wpl}(\mathrm{delTrans}(m, x_1, x_2), \psi)$
- $\mathrm{post}_2(\mathrm{delTrans}(m, x_1, x_2), \psi) = \psi \vee \mathrm{wpl}(\mathrm{addTrans}(m, x_1, x_2), \psi)$
- $\mathrm{post}_2(\mathrm{delTrans}(f, x), \psi) = \psi \vee \mathrm{wpl}(\mathrm{addTrans}(f, x, y), \psi)$
- $\mathrm{post}_2(\mathrm{addState}(x), \psi) = \mathrm{wpl}(\mathrm{delState}(x), \mathrm{wpl}(\mathrm{mvNom1}(x, y), \psi))$
- $\mathrm{post}_2(\mathrm{delState}(x), \psi) = \bigvee_{Y \subseteq A, Z \subseteq B} T_1(T_2(T_3^Y(T_4^Z(\psi))))$

where y is a fresh nominal, that is, a nominal that does not occur in ψ, A is the set of modality symbols occurred in ψ, and B is the set of nominals and propositional symbols occurred in ψ. Functions T_i ($i = 1, 2, 3, 4$) are defined as follows: $T_1(\psi) = \mathrm{wpl}(\mathrm{addState}(x), \psi)$. $T_2 = \mathrm{RD}\{\langle m \rangle \varphi \rightarrowtail \langle m \rangle \overline{\varphi} \vee (x \wedge \langle \mathbf{o} \rangle (y_m \wedge \overline{\varphi})) \vee (z_m \wedge @x\, \overline{\varphi}) \;\; (m \in \mathrm{MS})\}$, where y_m and z_m for $m \in \mathrm{MS}$ are fresh propositional symbols. $T_3^Y = \mathrm{RD}\{\langle m \rangle \varphi \rightarrowtail \langle m \rangle \overline{\varphi} \vee (x \wedge @x\, \overline{\varphi}) \;\; (m \in Y)\}$. $T_4^Z = \mathrm{RD}\{x' \rightarrowtail x \;\; (x' \in Z \cap \mathrm{Nom})\ ;\ p \rightarrowtail p \vee x \;\; (p \in Z \cap \mathrm{PS})\}$.

Let us check that the conditions hold for $\tau = \mathrm{mvNom1}(x_1, x_2)$ as an example. For the condition (2), assume $\mathcal{K}_2 \models \mathrm{post}(\tau, \psi)$. Let $\mathcal{K}_1$ be the result of $\mathrm{mvNom1}(x_1, y)$ applied to $\mathcal{K}_2$. Then $\mathcal{K}_1 \models \psi$ holds by Lemma 2 and $\mathcal{K}_2 \models \mathrm{post}_2(\tau, \psi)$, and $(\mathcal{K}_1, \mathcal{K}_2) \in \tau$ follows from the fact that $\mathcal{K}_2 \models \mathrm{post}_1(\tau)$. The condition (1) can be shown in a similar manner.

[*2] More precisely, a in $\mathrm{post}_1(\mathrm{addState}(x))$ should run over the set of atoms that occur on ψ because the length of the formula must be finite. Therefore $\mathrm{post}_1(\tau)$ depends also on ψ, but we prefer to conciseness.

```
                 { $@x\,\mathrm{EF}(a)$ }
y:= nil;         { $\psi$ }
while x != nil do    { $\psi \wedge \neg @x\,\mathbf{nil}$ }
  t := y;        { $(@t\,\mathrm{EU}(\neg x, a) \vee @x\,\mathrm{EU}(\neg t, a))$
                           $\wedge @x\,\neg\mathbf{nil} \wedge @t\,y$ }
  y := x;        { $\varphi_1 = (@t\,\mathrm{EU}(\neg y, a) \vee @y\,\mathrm{EU}(\neg t, a))$
                           $\wedge @y\,\neg\mathbf{nil} \wedge @y\,x$ }
  x := y.f;      { $\varphi_2 = (@t\,\mathrm{EU}(\neg y, a) \vee @y\,\mathrm{EU}(\neg t, a))$
                           $\wedge @y\,\langle f\rangle x$ }
                 { $\varphi_3 = (@t\,\mathrm{EU}(\neg y, a) \vee @y\,a$
                           $\vee @x\,\mathrm{EU}(\neg y, a)) \wedge @y\,\langle f\rangle x$ }
  y.f := t       { $\psi$ }
od               { $\psi \wedge @x\,\mathbf{nil}$ }    { $@y\,\mathrm{EF}(a)$ }
```

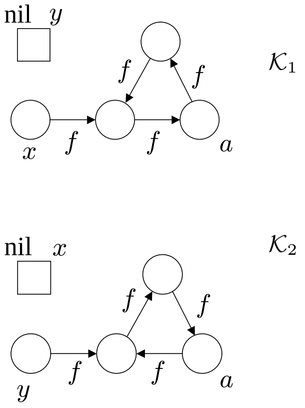

Fig. 1 A pointer-manipulating program

The conditions for other transformations can be checked similarly. The most complicated one is delState(x). The "reverse transformation" is defined as the composition of the following four transformations τ_i ($i = 1, \ldots, 4$). First, $\tau_1 = \mathrm{addState}(x)$. Second, τ_2 adds transitions for $m \in \mathrm{MS}$ from $L'(x)$ to states that satisfy y_m and from states that satisfy z_m to $L'(x)$. Third, τ_3 adds transitions for $m \in Y$ from x to itself. Fourth, τ_4 moves $L'(x')$ to the newly added state by τ_1 for $x' \in Z \cap \mathrm{Nom}$ and adds the state to $L(p)$ for $p \in Z \cap \mathrm{PS}$. For given Kripke structure $\mathcal{K}_2$, the resulting Kripke structure by transformation $\tau_4 \circ \tau_3 \circ \tau_2 \circ \tau_1$ can be regarded as $\mathcal{K}_1$ in the condition (2) since T_i ($i = 1, \ldots, 4$) in the definition of $\mathrm{post}_2(\mathrm{delState}(x), \psi)$ satisfies $T_i(\psi) = \mathrm{wpl}(\tau_i, \psi)$.

Thus, we establish the computation of the strongest postconditions. Unlike the weakest preconditions, if we work in the logic $\mathcal{L}'$, the strongest postconditions remain in $\mathcal{L}'$.

5 An Application to Pointer Manipulation

In this section, we illustrate how the results of the previous sections can be applied, by proving a property of a pointer-manipulating program.

Figure 1 shows the program written in a C-like language [9]. All variables (x, y, and t) are pointer-type and f is the name of a pointer-type field. The variables correspond to nominals and the field corresponds to a functional modality symbol.

Assume that the program is applied to $\mathcal{K}_1$ and it is transformed to $\mathcal{K}_2$. We verify that every state that is reachable from $L_1'(x)$ in $\mathcal{K}_1$ is reachable from $L_2'(y)$ in $\mathcal{K}_2$. Assertions are written in curly braces. The following abbreviations are used: $\mathrm{EU}(\varphi_1, \varphi_2) = \mu X(\varphi_2 \vee (\varphi_1 \wedge \langle f\rangle X))$, $\mathrm{EF}(\varphi) = \mathrm{EU}(\mathbf{true}, \varphi)$, and $\psi = @x\,\mathrm{EU}(\neg y, a) \vee @y\,\mathrm{EU}(\neg x, a)$.

We introduce a fresh nominal a and put formula $@x\,\mathrm{EF}(a)$ as the first assertion, which means "$L(a)$ is reachable from $L(x)$." We see formula $@y\,\mathrm{EF}(a)$ as the last assertion. Since a is fresh, this is what we need to deduce.

The weakest preconditions and/or the strongest postconditions are used to check that each step is correct. For example, the statement x:=y.f corresponds to the transformation $\tau = \mathrm{mvNom2}(x, y, f)$. In order to check the triple $\{\varphi_1\}\,$x := y.f$\,\{\varphi_2\}$, we use $\mathrm{post}(\tau, \varphi_1)$. The formula $@y\langle f\rangle x$ in φ_2 comes from $\mathrm{post}_1(\tau)$, $@t\,\mathrm{EU}(\neg y, a)$ and $@y\,\mathrm{EU}(\neg t, a)$ in φ_2 come from $\mathrm{post}_2(\tau, \varphi_1)$. The formula $@y\,\neg\mathbf{nil}$ can be used to assure that the program does not abort here, although we have not discussed the point in this paper. The next assertion φ_3 is justified by the fact that $\varphi_2 \to \varphi_3$ is a valid formula, which can be verified using appropriate decision procedures.

All formulas in this example are written in CTL (with nominals). However, we need general fixed-point operators to express more complex properties. For example, property "$L(a)$ is reachable from $L(x)$

by following f_1 and f_2 alternatively" is expressed by formula $@x\,\mu X(a \vee \langle f_1\rangle\langle f_2\rangle X)$, which cannot be expressed in CTL.

We have an experimental implementation [10] of a tool that verifies the properties of pointer-manipulating programs based on a similar technique described in this section. It calculates the weakest preconditions and has a sound (but incomplete) decision procedure for a sublogic of $\mathcal{L}$.

6 Conclusion and Future Work

We establish a method to compute pre- and post-conditions of formulas in variants of modal μ-calculus with regard to transformations of Kripke structures.

An obvious direction for future work is to implement the computation of pre- and post-conditions and combine them with decision procedures of the logic to build a verification system. As was already mentioned, it has been partially done, and we plan to extend it to fully cover the contents of this paper.

In this paper, we choose transformations of Kripke structures based on our intention to apply the results to analyze programs that manipulate single-valued pointers. Analyzing programs with multi-valued pointers should be attempted as future work.

Acknowledgments. This research was supported by research project "Solving the description explosion problem in verification by means of structure transformation" in CREST (Core Research for Evolution Science and Technology) program of Japan Science and Technology Agency.

The authors would like to thank Yoshiki Kinoshita of AIST and Masami Hagiya of the University of Tokyo for valuable comments and discussions.

References

[1] Takahashi, K., Hagiya, M.: Abstraction of graph transformation using temporal formulas. In: Supplemental Volume of the 2003 International Conference on Dependable Systems and Networks (DNS-2003). (2003) W–65 to W–66

[2] Hagiya, M., Takahashi, K., Yamamoto, M., Sato, T.: Analysis of synchronous and asynchronous cellular automata using abstraction by temporal logic. In: FLOPS2004: The Seventh Functional and Logic Programming Symposium. Volume 2998 of Lecture Notes in Computer Science. (2004) 7–21

[3] Blackburn, P.: Nominal tense logic. Notre Dame Journal of Formal Logic **34** (1993) 56–83

[4] Bonatti, P.A., Lutz, C., Murano, A., Vardi, M.Y.: The complexity of enriched μ-calculi. In Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., eds.: ICALP (2). Volume 4052 of Lecture Notes in Computer Science., Springer (2006) 540–551

[5] Bonatti, P.A., Peron, A.: On the undecidability of logics with converse, nominals, recursion and counting. Artificial Intelligence **158** (2004) 75–96

[6] Tanabe, Y., Takai, T., Sekizawa, T., Takahashi, K.: Preconditions of properties described in ctl for statements manipulating pointers. In: Supplemental Volume of the 2005 International Conference on Dependable Systems and Networks (DSN-2005). (2005) 228–234

[7] Sagiv, M., Reps, T., Wilhelm, R.: Parametric shape analysis via 3-valued logic. ACM Transactions on Programming Languages and Systems **24**(3) (2002) 217–298

[8] Distefano, D., O'Hearn, P.W., Yang, H.: A local shape analysis based on separation logic. In Hermanns, H., Palsberg, J., eds.: TACAS. Volume 3920 of Lecture Notes in Computer Science., Springer (2006) 287–302

[9] Kinoshita, Y., Nishizawa, K.: An algebraic semantics of predicate abstraction for PML. In: 24th Conference of Japan Society for Software Science and Technology. (2007)

[10] Sekizawa, T., Tanabe, Y., Yuasa, Y., Takahashi, K.: MLAT: A tool for heap analysis based on predicate abstraction by modal logic. In: IASTED International Conference on Software Engineering (SE 2008). (2008) 310–317

Appendix A Proofs

In the appendix, we prove Lemmas 1 and 2.

A formula is in *positive normal form (PNF)* if the negation symbol ($\neg$) only appears immediately before an atomic formula. The letter λ is used to stand for the fixed-point operator μ or ν. Thus, $\lambda X\varphi$ is either $\mu X\varphi$ or $\nu X\varphi$. The symbol $\{\}$ is used to stand for $\langle\rangle$ or $[]$. Thus, $\{\mathbf{o}\}$ is either $\langle\mathbf{o}\rangle$ or $[\mathbf{o}]$.

Lemma 6 Assume $\xi \in \mathcal{L}$ is in PNF and $\psi_0 = \{\mathbf{o}\}\psi$ is a subformula of ξ. Further assume $\lambda X\varphi$ is the innermost subformula of this form that contains ψ_0 as a subformula; that is, $\lambda X\varphi$ is a subformula of ξ, ψ_0 is a subformula of φ, but there is no subformula of φ in the form of $\lambda' Y\eta$ such that η contains ψ_0 as a subformula. Let $\varphi^{\mathrm{T}} = \varphi[\mathbf{true}/\psi_0]$ and $\varphi^{\mathrm{F}} = \varphi[\mathbf{false}/\psi_0]$, that is, φ^{T} is the formula obtained from φ by replacing all occurrences of ψ_0 with **true** and φ^{F} is similar. Let $\psi^{\mathrm{T}} = \psi[\lambda X\varphi^{\mathrm{T}}/X]$ and $\psi^{\mathrm{F}} = \psi[\lambda X\varphi^{\mathrm{F}}/X]$. Then if $\lambda = \mu$,

$$\mu X\varphi \equiv \{\mathbf{o}\}\psi^{\mathrm{F}} \to \mu X\varphi^{\mathrm{T}} \,; \mu X\varphi^{\mathrm{F}}$$

holds. Similarly if $\lambda = \nu$,

$$\nu X\varphi \equiv \{\mathbf{o}\}\psi^{\mathrm{T}} \to \nu X\varphi^{\mathrm{T}} \,; \nu X\varphi^{\mathrm{F}}$$

holds.

Proof. We only show the first half. The second half can be shown similarly.

Let $\mathcal{K} = (S, R, L, \mathrm{nil})$ be a Kripke structure and v a valuation for $\mathcal{K}$ and $s \in S$. We show:

(a) When $\mathcal{K}, v \models \{\mathbf{o}\}\psi^{\mathrm{F}}$, $\quad \mathcal{K}, v, s \models \mu X\varphi^{\mathrm{T}} \iff \mathcal{K}, v, s \models \mu X\varphi$.
(b) When $\mathcal{K}, v \not\models \{\mathbf{o}\}\psi^{\mathrm{F}}$, $\quad \mathcal{K}, v, s \models \mu X\varphi \iff \mathcal{K}, v, s \models \mu X\varphi^{\mathrm{F}}$.

Note that whether $\{\mathbf{o}\}\psi^{\mathrm{F}}$ is satisfied or not is independent from $s \in S$. In both cases the direction from right to left is clear since φ is in PNF, so we show the other direction.

For formula ξ, propositional variable X, and ordinal number α, we define $S(\xi, X, \alpha) \subseteq S$ as follows. We omit X and write $S(\xi, \alpha)$ if it is clear from the context.

- $S(\xi, 0) = \varnothing$
- $S(\xi, \alpha + 1) = [\![\varphi]\!]^{\mathcal{K}, v[X \mapsto S(\xi,\alpha)]}$
- $S(\xi, \alpha) = \bigcup_{\beta<\alpha} S(\xi, \beta) \qquad$ (if α is limit)

Let κ, κ^{T}, and κ^{F} be the ordinal numbers at which $S(\varphi, \cdot)$, $S(\varphi^{\mathrm{T}}, \cdot)$, and $S(\varphi^{\mathrm{F}}, \cdot)$, respectively, converges; that is, $S(\varphi, \kappa) = [\![\mu X\varphi]\!]$ holds and similarly for the others.

(a) Assume $\mathcal{K}, v \models \{\mathbf{o}\}\psi^{\mathrm{F}}$. We will show on induction α that

$$S(\varphi^{\mathrm{T}}, \alpha) \subseteq S(\varphi, \kappa^{\mathrm{F}} + \alpha). \tag{1}$$

Once it is established, by setting $\alpha = \kappa^{\mathrm{T}}$, we have $[\![\mu X\varphi^{\mathrm{T}}]\!] = S(\varphi^{\mathrm{T}}, \kappa^{\mathrm{T}}) \subseteq S(\varphi, \kappa^{\mathrm{F}} + \kappa^{\mathrm{T}}) \subseteq [\![\mu X\varphi]\!]$, which is to be proved in (a).

When $\alpha = 0$ or α is limit, (1) is clearly satisfied. For $\alpha + 1$, by induction hypothesis we have $S(\varphi^{\mathrm{T}}, \alpha+1) = [\![\varphi^{\mathrm{T}}]\!]^{v[X \mapsto S(\varphi^{\mathrm{T}},\alpha)]} \subseteq [\![\varphi^{\mathrm{T}}]\!]^{v[X \mapsto S(\varphi,\kappa^{\mathrm{F}}+\alpha)]}$. On the other hand, by definition, $S(\varphi, \kappa^{\mathrm{F}} + \alpha + 1) = [\![\varphi]\!]^{v[X \mapsto S(\varphi,\kappa^{\mathrm{F}}+\alpha)]}$. Therefore it is enough to show

$$\mathcal{K}, v[X \mapsto S(\varphi, \kappa^{\mathrm{F}} + \alpha)], s \models \varphi \leftrightarrow \varphi^{\mathrm{T}}. \tag{2}$$

From the assumption

$$\mathcal{K}, v \models \{\mathbf{o}\}\psi[\mu X\varphi^{\mathrm{F}}/X],$$

and therefore

$$\mathcal{K}, v[X \mapsto S(\varphi^{\mathrm{F}}, \kappa^{\mathrm{F}})] \models \{\mathbf{o}\}\psi.$$

Since $S(\varphi^{\mathrm{F}}, \kappa^{\mathrm{F}}) \subseteq S(\varphi, \kappa^{\mathrm{F}}) \subseteq S(\varphi, \kappa^{\mathrm{F}} + \alpha)$,

$$\mathcal{K}, v[X \mapsto S(\varphi, \kappa^{\mathrm{F}} + \alpha)] \models \{\mathbf{o}\}\psi,$$

which means

$$\mathcal{K}, v[X \mapsto S(\varphi, \kappa^{\mathrm{F}} + \alpha)] \models \{\mathbf{o}\}\psi \leftrightarrow \mathbf{true}.$$

Now (2) follows from the definition of φ^{T}.

(b) Assume $\mathcal{K}, v \not\models \{\mathbf{o}\}\psi^{\mathrm{F}}$. With similar argument as in part (a), it implies $\mathcal{K}, v[X \mapsto S(\varphi^{\mathrm{F}}, \kappa^{\mathrm{F}})], s \models \varphi \leftrightarrow \varphi^{\mathrm{F}}$.

We show by induction on α, $S(\varphi, \alpha) \subseteq S(\varphi^{\mathrm{F}}, \alpha)$. Cases for $\alpha = 0$ and limit ordinal α are trivial. For a successor ordinal, we have $S(\varphi, \alpha + 1) = [\![\varphi]\!]^{v[X \mapsto S(\varphi, \alpha)]} \subseteq [\![\varphi]\!]^{v[X \mapsto S(\varphi^{\mathrm{F}}, \alpha)]} = [\![\varphi^{\mathrm{F}}]\!]^{v[X \mapsto S(\varphi^{\mathrm{F}}, \alpha)]} = S(\varphi^{\mathrm{F}}, \alpha + 1)$.

Then, $[\![\mu X \varphi]\!] = S(\varphi, \kappa) \subseteq S(\varphi^{\mathrm{F}}, \kappa) \subseteq [\![\mu X \varphi^{\mathrm{F}}]\!]$ and we are done. ■

Lemma 7 Assume $\xi = \lambda X \varphi \in L$ and φ is FG-free. Then there is a FG-free formula that is equivalent to ξ

Proof. If ξ itself is not FG-free, pick a subformula $\psi_0 = \{\mathbf{o}\}\ \psi$ of φ such that the global modality does not appear in ψ. Then Lemma 6 can be applied and let ξ' be the resulting formula. Note that every subformula of ξ' in the form of $\lambda' Y \psi$, where $Y' \neq X$, is identical to a formula that exists in ξ. Also a subformula of ξ' whose principal operator is λX is either $\lambda X \varphi^{\mathrm{T}}$ or $\lambda X \varphi^{\mathrm{F}}$ and the number of occurrences of the global modality in φ^{T} and φ^{F} is strictly less than that in $\lambda X \varphi$. Therefore if we repeat this step for those subformulas whose principal operator is λX, the procedure completes after finitely many steps and we get a FG-free formula that is equivalent to ξ. ■

Proof of Lemma 1 Now we can prove the lemma by induction on the construction of the formula. In the case of μ and ν, we can use Lemma 7. The other cases are trivial. ■

Now assume that φ is a formula (that is not necessarily closed), ρ_1 and ρ_2 are valuations for $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, such that $\rho_1(X) \cap S_2 = \rho_2(X) \cap S_1$.

Lemma 8 (1) Assume τ is not in the form of addState(x). For any $s \in S_1 \cap S_2$, the following holds.

$$\mathcal{K}_1, \rho_1, s \models \mathrm{wpl}(\tau, \varphi) \iff \mathcal{K}_2, \rho_2, s \models \varphi$$

(2) Assume $x \in \mathrm{Nom}$ and $\tau = \mathrm{addState}(x)$. Let $\hat{s} = L_2(x)$, that is, $S_2 = S_1 \uplus \{\hat{s}\}$. We define a valuation ρ_1' for $\mathcal{K}_1$ by $\rho_1'(X) = S_1$ if $\hat{s} \in \rho_2(X)$ and $\rho_1'(X) = \varnothing$ if $\hat{s} \notin \rho_2(X)$. The following hold.

- Formulas $\mathrm{wpl}(\tau, \varphi)$ and $\mathrm{ns}(x, \varphi)$ are GV-free.
- $\mathcal{K}_1, \rho_1, s \models \mathrm{wpl}(\tau, \varphi) \iff \mathcal{K}_2, \rho_2, s \models \varphi$ holds for $s \in S_1$.
- $\mathcal{K}_2, \rho_2, \hat{s} \models \varphi \iff [\![\mathrm{ns}(x, \varphi)]\!]^{\mathcal{K}_1, \rho_1'} = S_1$.
- If $\mathcal{K}_2, \rho_2, \hat{s} \not\models \varphi \iff [\![\mathrm{ns}(x, \varphi)]\!]^{\mathcal{K}_1, \rho_1'} = \varnothing$.

Proof.

(1) The proof uses induction on the construction of φ. Since all of the cases can be shown in a straightforward manner, we only show one of the cases. The other cases can be proved similarly.

Let $\tau = \mathrm{addTrans}(x_1, x_2, m)$ and $\psi = \langle m \rangle \varphi$. In this case, we have $\mathrm{wpl}(\tau, \psi) = \langle m \rangle \mathrm{wpl}(\tau, \varphi) \vee (x_1 \wedge @x_2\, \mathrm{wpl}(\tau, \varphi))$. Assume $\mathcal{K}_1, \rho_1, s \models \mathrm{wpl}(\tau, \psi)$. If $\mathcal{K}_1, \rho_1, s \models \langle m \rangle \mathrm{wpl}(\tau, \varphi)$ holds, there is $s' \in S_1$ such that $(s, s') \in R_1(m)$ and $\mathcal{K}_1, \rho_1, s' \models \mathrm{wpl}(\tau, \varphi)$. By induction hypothesis $\mathcal{K}_2, \rho_2, s' \models \varphi$ and since $R_1(m) \subseteq R_2(m)$, we have $\mathcal{K}_2, \rho_2, s \models \psi$. If $\mathcal{K}_1, \rho_1, s \models x_1 \wedge @x_2, \mathrm{wpl}(\tau, \varphi)$, $s = L_1(x_1) = L_2(x_1)$ and $\mathcal{K}_1, \rho_1, L_1(x_2) \models \mathrm{wpl}(\tau, \varphi)$. By induction hypothesis $\mathcal{K}_2, \rho_2, L_2(x_2) \models \varphi$. Since $(L_2(x_1), L_2(x_2)) \in R_2(m)$, we have $\mathcal{K}_2, \rho_2, s \models \psi$. The other direction can be shown similarly.

(2) It is easy to check that $\mathrm{wpl}(\tau, \varphi)$ and $\mathrm{ns}(x, \varphi)$ are GV-free.

We next check $\mathcal{K}_1, \rho_1, s \models \mathrm{wpl}(\tau, \varphi) \iff \mathcal{K}_2, \rho_2, s \models \varphi$ holds for $s \in S_1$. There is no particular difficulty in checking cases other than $\langle \mathbf{o} \rangle \psi$. In the case of $\langle \mathbf{o} \rangle \psi$, note that since φ is a subformula of a GV-free formula, φ is GV-free. Therefore we can ignore the valuations, and this case can also be proved.

The remaining two items on ns can be shown in a similar manner. In the case of $\langle \mathbf{o} \rangle \psi$, use the fact that φ is GV-free and ψ does not contain free variables to ignore valuations, just as the case in $\mathrm{wpl}(\tau, \varphi)$. The other cases can be shown without difficulty. ■

Proof of Lemma 2 Clear from Lemma 8. ■

不動点演算子を持つ様相論理で表現された最弱事前条件と最強事後条件 (in English)
(算譜科学研究速報)
発行日：2008 年 4 月 25 日
編集・発行：独立行政法人 産業技術総合研究所 (システム検証研究センター)
同連絡先：〒560-0083 大阪府豊中市新千里西町 1-2-14 三井住友海上千里ビル 5F
TEL：06-4863-5025
e-mail：`informatics-inquiry@m.aist.go.jp`

Pre- and Post-conditions Expressed in Variants of the Modal μ-calculus
(Programming Science Technical Report)
25 April 2008
(Research Center for Verification and Semantics (CVS))
National Institute of Advanced Industrial Science and Technology (AIST)
5F Mitsui Sumitomo Kaijo Senri Bldg., 1-2-14, Shinsenrinishi-machi, Toyonaka, Osaka 560-0083 Japan
TEL：+81-6-4863-5025
e-mail：`informatics-inquiry@m.aist.go.jp`