AIST-PS-2006-003

A First-Order Extension of Modal μ-calculus

Keishi Okamoto

AIST, CVS

A First-Order Extension of Modal μ-calculus

Keishi Okamoto

AIST, CVS

A First-Order Extension of Modal μ-calculus

Keishi Okamoto

ABSTRACT. Modal μ-calculus is a modal logic with fixed-point operators and well-known in mathematics and computer scince. For example, many verification properties of a system are expressed by formulas of modal μ-calculus in computer science. However some verification properties of a system can not be expressed by a formula of modal μ-calculus as we will show later, and quantifiers of first-order logic are essentially required for expressing the verification properties. Therefore a first-order extension of modal μ-calculus is needed for expressing such verification properties. In this paper we introduce a first-order extension of modal μ-calculus and show that it is Σ^1_1-complete. Moreover we express some verification properties of a system by its formulas for showing its usefulness.

Keywords: modal μ-calculus, first-order extensions, non-recursive axiomatizability, formal methods, verification properties

1 Introduction

Formal methods are important in computer science and used for checking safety ciritical systems. In particular model checking is significant in formal methods. In model checking a verification property that is a property of a system required to be verified, is expressed by a formula of modal μ-calculus or its variants (linear temporal logic, computation tree logic) [11, 12]. Because they can express a safety property, a liveness property and so on. For example, mutual exclusion of two processes p_1 and p_2 (p_1 and p_2 will never be in their critical sections simultaneously) is expressed by the fomrula

$$\nu X.(\neg(CSP_1 \wedge CSP_2)) \wedge \Box X$$

of modal μ-calculus where the propositional variable CSP_1 (CSP_2) means that p_1 (respectively p_2) is in its critical section.

Consider a reactive system in which new processes is generated and spent processes vanish repeatedly. Since resources are always limited, the number of processes at a moment is bounded. But the number of whole processes, including all processes which will be generated in future, is unbouded because reactive systems will never stop. Then we have to check mutual exclusion of arbitrarily finitely many processes and it can not be expressed by a formula

of modal μ-calculus. Intuitively it requires the sequence

$$\nu X.(\neg \bigvee_{1 \leq i \neq j < \omega} (CSP_i \wedge CSP_j)) \wedge \Box X.$$

Of course this is possively a formula of an extension of modal μ-calculus which admits infinitely many disjuntions in its formula. But this property may also be expressed by the sequence

$$\nu X.(\neg \exists p_1.\exists p_2.p_1 \neq p_2 \wedge CS(p_1) \wedge CS(p_2)) \wedge \Box X$$

which is possibly a formula of a first-order extension of modal μ-calculus where $CS(x)$ is a predicate symbol meaning that a process x is in its critical section. Besides it is easier to understand the meaning of the second formula than the first one. Thus we decide to construct a first-order extension of modal μ-calculus.

Modal μ-calculus is a propositional modal logic with fixed-point operstors [12]. Then it is natural to add variables and quantifiers for objects to modal μ-calculus to construct a first-order extension of modal μ-calculus. On the other hand a first-order extension of modal μ-calculus can be considered as a fusion of modal μ-calculus and first-order modal logic [6] because it contains first-order modal logic and modal μ-calculus, and first-order modal logic contains variables and quantifiers for objects. So we fuse these two logics in section 2 and 3 and call it **first-order modal μ-calculus** (FOMμ) because modal μ-calculus is also called propositional modal μ-calculus. In section 4 we show that FOMμ is not recursively axiomatizable although modal μ-calculus and first-order modal logic are sound and complete for certain axiom systems. In section 5 we formalize a system to describe deadlocks those are important properties of a system in computer science. Then we express verification properties for deadlocks by formulas of FOMμ to show usefulness of FOMμ.

2 Syntax

As we stated in introdcution, we want the sequence

$$\nu X.(\neg \exists p_1.\exists p_2.p_1 \neq p_2 \wedge CS(p_1) \wedge CS(p_2)) \wedge \Box X$$

of symbols to have a meaning of mutual exclusion of arbitrarily finitely many processes. But it includes a predicate symbol CS, object variables p_1, p_2 and a quantifier $\exists$ of first-order modal logic and a propositional variable X and a fixed-point operator ν of modal μ-calculus simultaneously, so it is neither a formula of first-order modal logic nor that of modal μ-calculus. We have to merge syntax of modal μ-calculus and first-order modal logic into that of FOMμ to make the sequence a formula of FOMμ. Then formulas of modal μ-calculus and first-order modal logic are also formulas of FOMμ.

By the way first-order modal logic with constant symbols or function symbols makes troubles [5, 7, 10]. So we consider only a first-order extension of modal μ-calculus having neither constant symbols nor function symbols.

DEFINITION 1. A **signature** τ consists of predicate symbols $(P, Q, \cdots)$, each of which is of some fixed arity. A signature τ is also called a language or non-logical symbols.

Since first-order extensions of modal μ-calculus need variables for objects of first-order modal logic and variables for propositions of modal μ-calculus, so the set of variables of FOMμ also consists of two kinds of variables.

DEFINITION 2. A set $\mathcal{V}$ of variables is the union of a set $\mathcal{V}_o$ of infinitely many **object variables** $(x, y, \cdots)$ and a set $\mathcal{V}_p$ of infinitely many **propositional variables** $(X, Y, \cdots)$. We assume that $\mathcal{V}_o \cap \mathcal{V}_p = \emptyset$.

We note that propositional variables of FOMμ are different from predicate variables of second-order logic, so they have no arity.

Now we define the set of τ-formulas for a signature τ. Intuitively the set of atomic formulas of FOMμ consists of propositional variables of modal μ-calculus and atomic formulas of first-order logic. Then formulas of FOMμ are built up from these atomic formulas by the following rules. In particular a quantifer of first-order logic (a fixed-point operator of modal μ-calculus) can bind only object variables (respectively propositional variables).

DEFINITION 3. Let τ be a signature. We define the set of τ-formulas as follows:

1. if P is an n-ary predicate symbol in τ and $x_1, , \cdots, x_n$ are object variables then $P(x_1, \cdots, x_n)$ is a formula,
2. a propositional variable X is a formula,
3. if φ and ψ are formulas then $\neg\varphi, \varphi \vee \psi, \Box\varphi$ are formulas,
4. if $x \in \mathcal{V}_o$ and φ is a formula then $\forall x.\varphi$ is a formula,
5. if $X \in \mathcal{V}_p$ and φ is a formula in which any free occurrence of X must be within the scope of an even number of negation symbol (i.e. X is positive in φ) then $\mu X.\varphi$ is a formula.

We simply say a formula instead of a τ-formula if τ is clear from the context or not important and define freeness of a variable in a formula as usual. We also use the following usual abbreviations for readability.

- $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$,
- $\varphi \supset \psi = \neg\varphi \vee \psi$,
- $\exists x.\varphi = \neg\forall x.\neg\varphi$,
- $\nu X.\varphi = \neg\mu X.\neg\varphi[(\neg X)/X]$

where φ and ψ are formulas, $x \in \mathcal{V}_o, X \in \mathcal{V}_p$ and $\varphi[(\neg X)/X]$ denotes the syntactic substitution of a formula $\neg X$ for free occurrences of a propositional variable X in a formula φ.

We remark that fresh variables are required for a substitution of a formula for free occurrences of a propositional variable. For example

$$(\exists x.P(x) \vee X)[(\exists x.P(x) \vee X)/X] = \exists x.P(x) \vee (\exists x'.P(x') \vee X).$$

It is clear from the definition that formulas of modal μ-calculus and first-order modal logic are also formulas of FOMμ. Moreover the sequence

$$\nu X.(\neg\exists p_1.\exists p_2.p_1 \neq p_2 \wedge CS(p_1) \wedge CS(p_2)) \wedge \Box X$$

is a formula of FOMμ.

3 Semantics

In this section we define models, valuations and the truth values of formulas. Intuitively models of FOMμ are those of fist-order modal logic, valuations of FOMμ are fusions of those of first-order modal logic and modal μ-calculus and semantic functions of FOMμ are extensions of those of modal μ-calculus. We mostly adopt the terminology of [6].

A model of modal μ-calculus is a Kripke frame $\langle S, R\rangle$ where S is a set of states and R is a binary relation and every state can be considered as a model of propositional logic. A model of first-order extensions of modal μ-calculus requires that every state in the model can be considered as a model of first-order logic, so we adopt a model of first-order modal logic as that of FOMμ.

DEFINITION 4. A **frame** is a pair $\langle S, R\rangle$ where S is a set of states and R is a binary relation on S. R is called an accessibility relation or a transition relation. An **augmented frame** is a triple $\langle S, R, D\rangle$ where $\langle S, R\rangle$ is a frame and D is a non-empty set of objects. The set D of an augmented frame $\langle S, R, D\rangle$ is called a **domain** of $\langle S, R, D\rangle$.

To express properties by formulas of FOMμ as many as possible, we define the designation of a predicate symbol to be non-rigid and that of an object variable to be rigid. So an interpretation of FOMμ is the same as that of first-order modal logic.

DEFINITION 5. Let $\langle S, R, D\rangle$ be an augmented frame and τ a signature. An **interpretation** I in $\langle S, R, D\rangle$ is a function from the product $\tau \times S$ to the set $\bigcup_{n\in\omega} D^n$ such that $I(P, s)$ is an n-ary predicate on D for each n-ary predicate symbol P of τ and each state s of S.

Now we define a model of FOMμ, it is the same as a contant domain model of first-order modal logic.

DEFINITION 6. A **model** is a quadruple $\langle S, R, D, I\rangle$ where $\langle S, R, D\rangle$ is an augmented frame and I is an interpretation in $\langle S, R, D\rangle$. We say that a model $\langle S, R, D, I\rangle$ is **based on** the augmented frame $\langle S, R, D\rangle$.

In contrast to an interpretation, we define a valuation of FOMμ to be rigid, i.e. the designation of a free object variable does not depend on

states. Moreover FOMμ has object variables of first-order modal logic and propositional variables of modal μ-calculus, so we define a valuation as a mixture of valuations of them.

DEFINITION 7. Let $M = \langle S, R, D, I\rangle$ be a model. A **valuation** v in M is a function from the set $\mathcal{V}$ of all variables to the set $D \cup \mathcal{P}(S)$ such that

$$v(x) = \begin{cases} \text{an element } d \text{ of } D, & \text{if } x \in \mathcal{V}_o \\ \text{a subset } T \text{ of } S, & \text{if } x \in \mathcal{V}_p. \end{cases}$$

We prepare some notations for the truth value of a formula having a quantifier or a fixed-point operator. Let v be a valuation in a model $M = \langle S, R, D, I\rangle$, x an object variable and d an element of D. The valuation $v[d/x]$ in M is defined as follows:

$$v[d/x](y) = \begin{cases} d, & \text{if } y = x \\ v(y), & \text{if } y \neq x. \end{cases}$$

Let v be a valuation in M, X a propositional variable and T a subset of S. The valuation $v[T/X]$ is defined as follows:

$$v[T/X](Y) = \begin{cases} T, & \text{if } Y = X \\ v(Y), & \text{if } Y \neq X. \end{cases}$$

$v[d/x]$ and $v[T/X]$ are called **variants** of v. Since an object variable stands for an element of D, substituting an object variable for a subset of S is not allowed and vice versa.

In modal μ-calculus, the truth value of a formula ψ is defined by a function $[\![\ ]\!]$ so that, for a model $N = \langle S, R\rangle$ and a valuation v in N, $[\![\psi]\!]^N_v$ is the set $\{s \in S \mid \psi \text{ holds at } s \text{ in } N\}$ [12]. On the other hand, in first-order modal logic, the truth value of a formula φ is defined by a satisfaction relation $\models$ so that, for a model $M = \langle S, R, D, I\rangle$, a valuation v in M and a state $s \in S$,

$$M, s \models_v \varphi$$

means that φ is true at a state s [6]. Since a satisfaction relation $\models$ can be veiwed as a function $[\![\ ]\!]$ from the set of all formulas to $\mathcal{P}(S)$ such that

$$[\![\varphi]\!]^M_v = \{s \in S \mid M, s \models_v \varphi\},$$

we can merge a satisfaction relation of first-order modal logic and a function of modal μ-calculus into one. So we extend the funcion $[\![\ ]\!]$ of modal μ-calculus to evaluate all formulas of FOMμ because it is easy to define the truth value of a formula with a fixed-point operator with the function $[\![\ ]\!]$ of modal μ-calculus.

DEFINITION 8. Let $M = \langle S, R, D, I\rangle$ be a model and v a valuation in M. We define a function $[\![\ ]\!]^M_v$ from the set of all formulas to $\mathcal{P}(S)$ as follows:

1. $[\![P(x_1, \cdots, x_n)]\!]^M_v = \{s \mid (v(x_1), \cdots, x(x_n)) \in I(P, s)\}$ where P is an n-array predicate symbol and $x_1, \cdots, x_n \in \mathcal{V}_o$,
2. $[\![X]\!]^M_v = v(X)$ where $X \in \mathcal{V}_p$,
3. $[\![\neg\varphi]\!]^M_v = S \setminus [\![\varphi]\!]^M_v$,
4. $[\![\varphi \wedge \psi]\!]^M_v = [\![\varphi]\!]^M_v \cap [\![\psi]\!]^M_v$,
5. $[\![\forall x.\varphi]\!]^M_v = \bigcap_{d \in D} [\![\varphi]\!]^M_{v[d/x]}$,
6. $[\![\Box\varphi]\!]^M_v = \{s \in S \mid$ for all $t \in S$, if $R(s,t)$ holds then $t \in [\![\varphi]\!]^M_v\}$,
7. $[\![\mu X.\varphi]\!]^M_v = \bigcap\{T \subseteq S \mid [\![\varphi]\!]^M_{v[T/X]} \subseteq T\}$.

We simply write $[\![\varphi]\!]_v$ for $[\![\varphi]\!]^M_v$ if M is clear from the context.

We can also recover the satisfaction relation $\models$ from a function $[\![\ ]\!]$.

DEFINITION 9. Let $M = \langle S, R, D, I\rangle$ be a model, v a valuation in M and $s \in S$.

- $M, s \models_v \varphi$ if $s \in [\![\varphi]\!]^M_v$, and say that φ is **true** at s in M under v,
- $M \models \varphi$ if $M, s' \models_{v'} \varphi$ for any state $s' \in S$ and any valuation v' in M, and say that φ is **valid in** M,
- $\models \varphi$ if $M' \models \varphi$ for any model M', and say that φ is **valid**.

3.1 Completeness Results for Fragments of FOMμ

First-order modal logic is a fragment of FOMμ without a fixed-point operator and modal μ-calculus is a fragment of FOMμ without a quantifier because syntax and semantics of FOMμ are exntesions of those of modal μ-calculus and first-order modal logic. Moreover it is well-known that they are sound and complete for certain axiom systems [10, 13].

We also have a fragment of FOMμ without a modal operator. Since the fragment does not have modal operators, it is natural to assume that every frame of the fragment is a frame $\langle S, R\rangle$ of FOMμ where S is a singleton set and R is empty. Tatsuta showed that it is sound and complete for the well-known axiom system of first-order logic with the following inferrence rules:

$$\frac{\varphi}{\mu X.\varphi}\ \mu I' \qquad \frac{\mu X.\varphi}{\varphi}\ \mu E'$$

X is positive in φ in $\mu I'$.

by the following claim.

CLAIM 10 (Tatsuta). Let FOμ be the fragment of FOMμ without modal operators such that its frame $\langle S, R\rangle$ is a frame of FOMμ where S is a singleton set and R is empty, and other definitions are the same as FOMμ. Suppose that a propositional variable X is positive in a formula φ of FOμ. Then φ is valid in FOμ if and only if $\mu X.\varphi$ is valid in FOμ.

Proof. Let $M = \langle S, R, D, I\rangle$ be a model of FOμ. We show that φ is valid in M if and only if $\mu X.\varphi$ is valid in M.

($\Rightarrow$) Suppose that φ is valid in M hence $[\![\varphi]\!]_{v[\emptyset/X]} = S$ for any valuation v in M. By monotonicity of the function which takes a subset T of S to $[\![\varphi]\!]_{v[T/X]}$, $[\![\varphi]\!]_{v[T'/X]} = S$ for any subset T' of S and any valuation v in M. Thus $\{T' \mid [\![\varphi]\!]_{v[T'/X]} \subseteq T'\} = \{S\}$ for any valuation v in M, so $\mu X.\varphi$ is valid in M.

($\Leftarrow$) Suppose that $\mu X.\varphi$ is valid in M hence $[\![\mu X.\varphi]\!]_v = S$ for any valuation v in M. Since S is a singleton set,

$$[\![\mu X.\varphi]\!]_v = \begin{cases} \emptyset & \text{if } [\![\varphi]\!]_{v[\emptyset/X]} = \emptyset, \\ S & \text{otherwise } ([\![\varphi]\!]_{v[\emptyset/X]} = S). \end{cases}$$

for any valuation v in M. Then $[\![\varphi]\!]_{v[\emptyset/X]} = S$ for any valuation v in M. By monotonicity of the function which takes a subset T of S to $[\![\varphi]\!]_{v[T/X]}$, $[\![\varphi]\!]_{v[T'/X]} = S$ for any subset T' of S and any valuation v in M. Thus φ is valid in M. ■

The above three fragments of FOMμ are sound and complete for certain axiom systems, so we conjectured that FOMμ itself is also sound and complete for an axiom system. But this conjecture is false as we will show in the next section.

4 FOMμ is not recursively axiomatizable

Tiling problems are used to show undecidability or non-recursively axiomatizability of modal logics. For example, undecidability of a variant of deterministic propositional dynamic logic, called KRA, is proved by reducing the $\omega \times \omega$ tiling problem, that is Π^0_1-complete, to a satisfiability problem of it [1]. Non-recursively axiomatizability of KR[$*$] and logics of common knowledge are proved by reducing the $\omega \times \omega$ recurrent tiling problem, that is Σ^1_1-complete, to satisfiability problems of them [1, 14]. In this section we reduce the $\omega \times \omega$ recurrent tiling problem to a satisfiability problem of FOMμ for showing that it is not recursively axiomatizable.

Intuitively a tile is a 1×1 square, fixed orientation, each side of which has a color. Formally a **tile** T is a quadruple $\langle c_1, c_2, c_3, c_4\rangle$ where c_1, c_2, c_3, c_4 are in a set of colors with four functions r (right), l (left), u (up), d (down) such that $r(T) = c_1, l(T) = c_2, u(T) = c_3, d(T) = c_4$. We say that two tiles are equivalent if each side of them has the same color. A **tile type** t is an equivalence class of tiles up to this equivalence relation. Now we introduce the $\omega \times \omega$ recurrent tiling problem.

DEFINITION 11 (The $\omega \times \omega$ recurrent tiling problem [1]). Given a finite set $\mathcal{T}$ of tile types including some distinguished tile type t_0, can $\mathcal{T}$ tile $\omega \times \omega$ in such a way that adjacent tiles have the same color on the common sides, and that t_0 occurs infinitely often in the first row?

Let $\mathcal{T}$ be a set of finitely many tile types $t_0, t_1, \ldots, t_n$. To reduce the $\omega \times \omega$ recurrent tiling problem of $\mathcal{T}$ to a satisfiability problem of FOMμ, we construct a formula φ that is the conjunction of

- φ_{grid} whose intended meaning is that there is an $\omega \times \omega$ grid in $S \times D$ for a model $\langle S, R, D, I\rangle$,
- $\varphi_{\mathcal{T}}$ whose intended meaning is that there is a tiling of $\mathcal{T}$ on the $\omega \times \omega$ grid defined by φ_{grid} and
- φ_{rec} whose intended meaning is that t_0 occurs infinitely often in the first row of the $\omega \times \omega$ tiling defined by $\varphi_{\mathcal{T}}$.

and show that the satisfiability of φ is equivalent to an $\omega \times \omega$ recurrent tiling of $\mathcal{T}$.

We mention meanings of two formulas to construct φ. Let θ be a formula, $M = \langle S, R, D, I\rangle$ a model, v a valuation in M and $s \in S$. Since semantics of FOMμ is an extension of those of modal μ-calculus, the statement

$$M, s \models_v \nu X.\theta \wedge \Box X$$

means that θ holds at any reachable states from s and the statement

$$M, s \models_v \nu X.\mu Y.((\theta \vee \Diamond Y) \wedge \Diamond X)$$

means that there is a path in $\langle S, R\rangle$ in which θ holds infinitely often. The first one is for a grid and a tiling of $\mathcal{T}$ on the grid. The second is for recurrence of the tile type t_0.

Now we construct the formula φ. Let $\tau_{\mathcal{T}}$ be a signature consisting of a binary predicate symbol R' and unary predicate symbols T_i for each tile type t_i in $\mathcal{T}$.

φ_{grid} is the conjunction of

g1 $\nu X.(\Diamond(\forall x.x = x)) \wedge \Box X$,

g2 $\forall x.\exists y.R'(x, y)$,

g3 $\forall x, y.(R'(x, y) \supset (\nu X.R'(x, y) \wedge \Box X))$ and

g4 $\forall x, y.(\neg R'(x, y) \supset (\nu X.\neg R'(x, y) \wedge \Box X))$.

Let $\langle S, R, D, I\rangle$ be a model and suppose that φ_{grid} is true at a state s in S. Then g1 requires that $\langle S_0, R_0\rangle$ is serial for the set S_0 of all reachable states from s and the restriction R_0 of R to S_0. g2 requires that $\langle D, R'\rangle$ is serial at s. g3 and g4 require that $I(R', s) = I(R', t)$ for any reachable state t from s.

$\varphi_{\mathcal{T}}$ is the conjunction of

t1 $\nu X.(\forall x. \bigvee_{0\leq i\leq n} T_i(x)) \wedge \Box X$,

t2 $\nu X.(\forall x. \bigwedge_{0\leq i\neq j\leq n}(T_i(x) \supset \neg T_j(x))) \wedge \Box X$,

t3 $\nu X.(\forall x.\forall y.(T_i(x) \wedge R'(x,y) \supset \bigvee_{u(t_i)=d(t_j)} T_j(y))) \wedge \Box X$ $(0 \leq i \leq n)$ and

t4 $\nu X.(\forall x.(T_i(x) \supset \Box \bigvee_{r(t_i)=l(t_j)} T_j(x))) \wedge \Box X$ $(0 \leq i \leq n)$.

Let $M = \langle S, R, D, I\rangle$ be a model and suppose that $\varphi_{\mathcal{T}}$ is true at a state s. By t1,

$$M, t \models \bigvee_{0\leq i\leq n} T_i(d)$$

for any element d in D and any reachable state t from s. Intuitively this requires that every pair (t, d) has a tile type. Similarly t1 and t2 require that every pair has a unique tile type. Moreover t3 requires that tiles on pairs (t, d) and (t, d') for a reachable state t from s and elements d, d' in D with $R'(d, d')$, i.e. vertically adjacent tiles, have the same color on the common sides. t4 requires that horizontally adjacent tiles have the same color on the common sides by a similar argument.

φ_{rec} is the formula

$$\exists x.(\nu X.\mu Y.(T_0(x) \vee \Diamond Y) \wedge \Diamond X).$$

requiring that there are an element d of D and a path in $\langle S, R\rangle$ such that $T_0(d)$ occurs infinitely often in the path as we mentioned.

Now we show the equivalence of an $\omega \times \omega$ recurrent tiling of $\mathcal{T}$ and satisfiability of φ $(= \varphi_{grid} \wedge \varphi_{\mathcal{T}} \wedge \varphi_{rec})$.

PROPOSITION 12. *φ is satisfiable if and only if there is an $\omega \times \omega$ recurrent tiling of $\mathcal{T}$ hence the $\omega \times \omega$ recurrent tiling problem is reducible to a satisfiability problem of FOMμ. Then the satisfiability problem of FOMμ is Σ^1_1-complete, so FOMμ is not recursively axiomatizable.*

Proof. We remark that the truth value of φ does not depend on a valuation because it does not have any free variables. So we omit a valuation.

($\Leftarrow$) Suppose that there is an $\omega \times \omega$ recurrent tiling of $\mathcal{T}$. Consider the model $M = \langle \omega, R, \omega, I\rangle$ where

- $R = \{(m, m+1) \mid m \in \omega\}$,
- $I(R', m) = \{(n, n+1) \mid n \in \omega\}$ for all $m \in \omega$ and
- $I(T_i, m) = \{n \in \omega \mid (m, n)$ has a tile type $t_i\}$ for each T_i $(0 \leq i \leq n)$ and $m \in \omega$.

Since $\langle \omega, R\rangle$ ($\langle \omega, R'\rangle$) is serial, g1 (respectively g2) holds at the state 0. g3 and g4 also hold at 0 because $I(R', 0) = I(R', m)$ for any $m \in \omega$. Every pair (m, n) of a state $m \in \omega$ and an element $n \in \omega$ has a unique tile type,

so t1 and t2 hold at 0 by the interpretation I. Since, for every $m, n \in \omega$, vertically adjacent tiles on pairs (m, n) and $(m, n+1)$ in the $\omega \times \omega$ recurrent tiling have the same color on the common sides,

$$M, m \models T_i(n) \wedge R'(n, n+1) \supset \bigvee_{u(t_i)=d(t_j)} T_j(n+1)$$

holds for any T_i hence t3 holds. By a similar argument, t4 also holds. Finally $\{m \mid M, m \models T_0(0)\}$ is an infinite set by the definition of I. So

$$M, 0 \models \nu X.\mu Y.((T_0(0) \vee \Diamond Y) \wedge \Diamond X)$$

holds hence φ_{rec} holds at 0.

($\Rightarrow$) Suppose that $M, s_0 \models \varphi$ for a model $M = \langle S, R, D, I\rangle$ and $s_0 \in S$. By φ_{rec}, there is an element d_0 of D and a path $\pi_S = \{s_m \in S \mid R(s_m, s_{m+1}), m \in \omega\}$ from s_0 in which $T_0(d_0)$ holds infinitely often. On the other hand, by g2, there is an infinite path $\pi_D = \{d_n \in D \mid R'(d_n, d_{n+1}), n \in \omega\}$ from d_0. Moreover by g3 and g4, $I(R', s_0) = I(R', s_m)$ for all $s_m \in \pi_S$ hence $\pi_S \times \pi_D$ forms an $\omega \times \omega$ grid such that $T_0(d_0)$ holds infinitely often in its fisrt row. Then we show that the function $f\colon \pi_S \times \pi_D \to \mathcal{T}$ which takes (s_m, d_n) to t_i when $M, s_m \models T_i(d_n)$ gives an $\omega \times \omega$ recurrent tiling of $\mathcal{T}$. Recurrence of t_0 in its first row is clear, so we show the rest.

By t1 and t2,

$$M, s \models \bigvee_{0 \le i \le n} T_i(d), \quad M, s \models \bigwedge_{0 \le i \ne j \le n} (T_i(d) \supset \neg T_j(d))$$

for every pair (s, d) of a reachable state $s \in S$ from s_0 and an element $d \in D$ hence (s, d) has a unique tile type. By t4,

$$M, s \models T_i(d) \supset \Box \bigvee_{r(t_i)=l(t_j)} T_j(d)$$

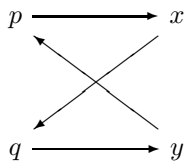
holds for any reachable state $s \in S$ from s_0 and any element $d \in D$. In particular horizontally adjacent tiles on (s_m, d_n) and (s_{m+1}, d_n) in $\pi_S \times \pi_D$ have the same color on the common sides. By a similar argument, vertically adjacent tiles on (s_i, d_j) and (s_i, d_{j+1}) in $\pi_S \times \pi_D$ have the same color on the common sides by t3. ■

5 An Application of FOMμ to Deadlocks

In this section we formalize resource allocation system for deadlocks in FOMμ and express verification properties for deadlocks by formulas of FOMμ. A deadlock is a situation in which no process can act. For example, there are one pencil, one ruler and two persons p_1 and p_2 who want to use the pencil and the ruler to draw a straight line. If p_1 takes the pencil and p_2 takes the ruler then p_1 has to wait for the ruler held by p_2 and p_2 has to wait for the pencil held by p_1 hence a deadlock occurs. Deadlocks also occur in various resource allocation systems so they are a considerable problem in computer science [3, 8].

5.1 Modeling Resource Allocation Systems

Deadlocks are problems of resource allocation and a resource allocation graph is used for describing a situation of resource allocation at a moment. A resource allocation graph is a bipartite directed graph such that a class of vertices is a set of processes and the other is a set of resources. In a resource allocation graph, an edge from a process p to a resource x means that p is requiring x and an edge from a resource y to a process q means that y is allocated to q. Then the graph



processes resources

represents a situation in which

1. a process p is requesting a resource x,
2. x is allocated to a process q,
3. q is requesting a resource y and
4. y is allocated to p. (so a deadlock occurs.)

A resource allocation graph is described in first-order logic. A signature τ_0 of first-order logic for resource allocation graphs consists of

- a unary predicate symbol $Proc(x)$ (x is a process),
- a unary predicate symbol $Res(x)$ (x is a resource) and
- a binary predicate symbol $E(x, y)$ (there is an edge from x to y).

Moreover we use two abbreviations for readability.

- $Rq(p, x) = Proc(p) \wedge Res(x) \wedge E(p, x)$ (p is requesting x)
- $Al(p, x) = Proc(p) \wedge Res(x) \wedge E(x, p)$ (x is allocated to p)

Then the above graph is expressed by the formula

$$Rq(p, x) \wedge Al(q, x) \wedge Rq(q, y) \wedge Al(p, y).$$

We call a system managing resource allocation a resource allocation system. For example, a memory allocation system in a computer is a resource allocation system. Then a snapshot of a resource allocation system is a situation of resource allocation, i.e. a resource allocation graph and a change in a resource allocation system is an action of a process, i.e. a transition

between resource allocation graphs. Thus the structure of a resource allocation system is captured by a model of FOMμ.

The model of FOMμ for a resource allocation system $\mathcal{S}$ is a model $\langle S, R, D, I\rangle$ where D is the set of all processes and resources in $\mathcal{S}$, S is the set of all resource allocation graphs over D representing situations in $\mathcal{S}$, R is the set of all transitions on S representing actions of processes in $\mathcal{S}$ and I is the canonical interpretation in $\langle S, R, D\rangle$ for $\mathcal{S}$.

We mention that D may be an infinite set even if there are at most finitely many processes and resources at a moment in $\mathcal{S}$. Because new processes are generated and spent processes vanish repeatedly in $\mathcal{S}$.

A signature τ of FOMμ for resource allocation systems is the same as the signature τ_0 of first-order logic for resource allocation graphs. But interpretations of them are different from those of τ_0 because interpretations of FOMμ depend on states. By virtue of this, we can express dynamic generarion of processes. For example, the formula

$$\exists x.(\neg Proc(x) \wedge \Diamond Proc(x))$$

says that there is a process such that it does not exist now but it will be generated at a next state.

5.2 Banker's algorithm

We express a verification property for deadlock avoidance, called banker's algorithm [4]. The algorithm investigates whether a state (a situation) is safe, i.e. there is a sequence of states from the state along which the resources (money) requested by processes (customers) can eventually be allocated. So deadlocks are avoidable at a state if the state is safe. To find such a sequence, the algorithm investigates whether a process is reducible at a state, i.e. there is a sequence of states from the state along which a system (a banker) can keep satisfying requests of the process successively until all its requests are satisfied. If the process is reducible then the algorithm investigate the other processes as if all the resources allocated to the last process have released. The algorithm find a total sequence for safty of the state if all processes can be reducible in certain order.

More precisely, for a model $M = \langle S, R, D, I\rangle$, the algorithm says that a state s_0^0 is safe if there are

- a set $\{p_i \in D \mid 0 \leq i \leq m\}$ of processes,
- a set $\{x_i^{j_i} \in D \mid 0 \leq i \leq m, 0 \leq j_i \leq n_i - 1\}$ of resources and
- a path $\{s_i^{j_i} \in S \mid 0 \leq i \leq m, 0 \leq j_i \leq n_i\}$ of states

such that

1. $s_i^{j_i} R s_i^{j_i+1} \quad (0 \leq i \leq m, 0 \leq j_i < n_i)$,
2. $s_i^{n_i} R s_{i+1}^1 \quad (0 \leq i < m)$,

3. $M, s_0^0 \models Rq(p_0, x_0^0)$,
4. $M, s_i^0 \models \Omega(p_{i-1}) \wedge Rq(p_i, x_i^0) \quad (1 \leq i \leq m)$,
5. $M, s_i^{j_i} \models Rq(p_i, x_i^{j_i}) \wedge Al(p_i, x_i^{j_i - 1}) \quad (0 \leq i \leq m, 1 \leq j_i \leq n_i - 1)$,
6. $M, s_i^{n_i} \models Al(p_i, x_i^{n_i - 1}) \wedge \Omega'(p_i) \quad (0 \leq i \leq m)$ and
7. $M, s_m^{n_m} \models \Diamond(\Omega(p_m) \wedge \Omega)$

where

- $\Omega'(p) = \neg\exists x.Rq(p, x)$,
- $\Omega(p) = \neg\exists x.Rq(p, x) \vee Al(p, x)$ and
- $\Omega = \neg\exists p.\exists x.Rq(p, x) \vee Al(p, x)$.

Intuitively, by (3), (4), (5) and (6), the sets $\{x_i^{j_i} \mid 0 \leq j_i \leq n_i - 1\}$ and $\{s_i^{j_i} \mid 0 \leq j_i \leq n_i\}$ shows that p_i will be reducible for $0 \leq i \leq m$. In addition to this, all processes can be reducible by (7), so the state s_0^0 is safe.

Before expressing banker's algorithm, we express banker's algorithm for an elementary case in which only one process requires resources. Let $M = \langle S, R, D, I\rangle$ be a model and v a valuation in M. Consider the formula

$$\Phi_0'(p) : \mu Y.(\Omega'(p) \wedge \Diamond(\Omega(p) \wedge \Omega)) \vee (\exists x.Rq(p, x) \wedge \Diamond(Al(p, x) \wedge Y)).$$

Suppose $M, s_0 \models_v \Phi_0'(p)$ for a state $s_0 \in S$. Then there are a natural number n, a set $\{x_i \in D \mid 0 \leq i < n\}$ of resources and a path $\{s_j \in S \mid 0 \leq j \leq n\}$ of states such that

- $M, s_0 \models_v Rq(p, x_0)$,
- $M, s_i \models_v Rq(p, x_i) \wedge Al(p, x_{i-1}) \quad (0 < i < n)$ and
- $M, s_n \models_v Al(p, x_{n-1}) \wedge \Omega'(p) \wedge \Diamond(\Omega(p) \wedge \Omega)$.

(We require only that $\Omega'(p) \wedge \Diamond(\Omega(p) \wedge \Omega)$ holds at s_0 when $n = 0$.) Thus the formula $\Phi_0'(p)$ expresses the algorithm for the elementary case.

Now we construct a formula Φ expressing banker's algorithm, i.e. for a model $M = \langle S, R, D, I\rangle$ and a state s in S, Φ holds at s in M if and only if there are a set of processes, a set of resources and a sequence of states from s satisfying the conditions (1)-(6) in M.

By an analogy with elementary case,

Φ is the disjunction of the formula Ω and the least fixed-point of a sequence $\{\Phi_i \mid i \in \omega\}$ of formulas such that

$$\Phi_0 : \exists p.\mu Y.(\Omega'(p) \wedge \Diamond(\Omega(p) \wedge \Omega)) \vee (\exists x.Rq(p, x) \wedge \Diamond(Al(p, x) \wedge Y)) \text{ and}$$

$$\Phi_{n+1} : \exists p.\mu Y.(\Omega'(p) \wedge \Diamond(\Omega(p) \wedge (\Omega \vee \Phi_n))) \vee (\exists x.Rq(p, x) \wedge \Diamond(Al(p, x) \wedge Y)).$$

So Φ is the formula

$$\Omega \vee \mu X.\exists p.\mu Y.\Psi(p, X, Y) \quad \text{(Banker's algorithm)}$$

where $\Psi(p, X, Y)$ is the formula

$$(\Omega'(p) \wedge (\Diamond\Omega(p) \wedge (\Omega \vee X))) \vee (\exists x.Rq(p, x) \wedge \Diamond(Al(p, x) \wedge Y)).$$

6 Future work

We have constructed a first-order extension of modal μ-calculus FOMμ. Because of its high expressive power, it lose decidability. So it is worth while considering decidable fragments of FOMμ, as certain fragments of first-order temporal logic are decidable [9].

On the other hand we are formalizing systems for verification and expressing verification properties as in section 5. For example, it is well-known that Coffman conditions (mutual exclusion, hold and wait, no preemption and circular wait) are necessary for deadlocks. They are also sufficient for deadlocks under certain assumptions [3, 8]. So it is worth while expressing deadlocks and these conditions by formulas of FOMμ to check whether a system has a deadlock. Moreover, if we have a powerful and sound axiom system for FOMμ, we may prove a deadlock from these conditions in the axiom system.

Acknowledgements

I wish to express my thanks to members of Research Center for Verification and Semantics (CVS) and Izumi Takeuti for valuable discussions and suggestions, and especially to Yoshiki Kinoshita for showing importance of a first-order extension of modal μ-calculus in mathematics and computer science. I am also grateful to Makoto Tatsuta for valuable suggestions.

BIBLIOGRAPHY

[1] Patrick Blackburn, Maarten de Rijke and Yde Venema, *Modal Logic*, Cambridge Tracts in Theoretical Computer Science 53

[2] Julian Charles Bradfield, *Verifying Temporal Properties of Systems*, Progress in Theoretical Computer Science, Birkhäuser

[3] E.G.Coffman, JR., M.J.Elphick and A.Shoshani, *System Deadlocks*, Rapport Technique EWD-123, Computing Surveys, Vol. 3, No. 2, June 1971

[4] Edsger W. Dijkstra, *Cooperating sequential processes*, Rapport Technique EWD-123, Technological University, Eindhoven, the Netherlands

[5] Melvin Fitting, *Basic Modal Logics*, Handbook of Logic in Artificial Intelligence and Logic Programming Volume 1 Logical Foundation pp 365-448, Oxford Science Publications

[6] Melvin Fitting and Richard L. Mendelsohn, *First-order Modal Logic*, Synthese Library/volume 277, Kluwer Academic Publications

[7] Ronald Fagin, Joseph Y. Halpern, Yoram Moses and Moshe Y. Vardi, *Reasoning About Knowledge*, The MIT Press

[8] J.W.Havender *Avoiding deadlock in multitasking systems*, IBM Systems Journal 2 (1968), 74-84

[9] Ian Hodkinson, Frank Wolter and Michael Zakharyaschev *Decidable fragments of first-order temporal logics*, Annals of Pure and Applied Logic 106 (2000) 85-134

[10] G.E.Hughes and M.J.Cresswell, *A new introduction to Modal Logic*, Routledge 1996
[11] M.Huth and M.Ryan, *Logic in Computer Science: Modeling and Reasoning About Systems*, Cambridge University Press 2004
[12] Dexter Kozen, *Results on the Propositional μ-calculus*, Theoretical Computer Science 27 (1983) 333-354
[13] Igor Walukiewicz, *Completeness of Kozens axiomatisation of propositional mu-calculus*, LICS 95, Information and Computation 157, pp.142–182 2000
[14] Frank Wolter, *First order common knowledge logics*, Studia Logica 65 (2000) pp 249-271

Keishi Okamoto
Research Center for Verification and Semantics (CVS),
National Institute of Advanced Industrial Science and Technology (AIST)
5th Floor, Mitsui Sumitomo Kaijo Senri Bldg.,
1-2-14 Shin-Senri Nishi, Toyonaka, Osaka, 560-0083 Japan
E-mail: keishi-okamoto@aist.go.jp

A First-Order Extension of Modal μ-calculus
(算譜科学研究速報)
発行日：2006 年 4 月 20 日
編集・発行：独立行政法人産業技術総合研究所関西センター尼崎事業所
システム検証研究センター
同連絡先：〒661-0974 兵庫県尼崎市若王寺 3-11-46
e-mail：informatics-inquiry@m.aist.go.jp

A First-Order Extension of Modal μ-calculus
(Programming Science Technical Report)
Apr. 20, 2006
Research Center for Verification and Semantics (CVS)
AIST Kansai, Amagasaki Site
National Institute of Advanced Industrial Science and Technology (AIST)
3-11-46 Nakouji, Amagasaki, Hyogo, 661-0974, Japan
e-mail: informatics- inquiry@m.aist.go.jp