Multi-Valued Modal Fixed Point Logics for Model Checking

Koki NISHIZAWA^{†a)}, Nonmember

SUMMARY In this paper, I will show how multi-valued logics are used for model checking. Model checking is an automatic technique to analyze correctness of hardware and software systems. A model checker is based on a temporal logic or a modal fixed point logic. That is to say, a system to be checked is formalized as a Kripke model, a property to be satisfied by the system is formalized as a temporal formula or a modal formula, and the model checker checks that the Kripke model satisfies the formula. Although most existing model checkers are based on 2-valued logics, recently new attempts have been made to extend the underlying logics of model checkers to multi-valued logics. I will summarize these new results.
key words: fixed point logic, De Morgan algebra, Heyting algebra, min-plus algebra

1. Introduction

In order to discover bugs of hardware or software, various formal methods have been studied. *Model checking* [6] is the most successful as the technique to automatically verify transition systems modelling hardware and software against temporal formulas expressing specifications. The specifications for model checking are described in modal logics, temporal logics, and modal fixed point logics [4], [16], [17], [22], [27]. Until the 1990s, the logics used for model checking were based on only two truth values, i.e., ‘true’ and ‘false’. However, in the 2000s, multi-valued logics for model checking are studied extensively. This paper introduces their logics.

First, we introduce the ordinary logics for model checking. The system to be verified is formalized as a Kripke structure. A *Kripke structure* consists of a set S , a subset $\rightarrow$ of $S \times S$, and a subset ρ of $S \times \mathbf{Atom}$, where $\mathbf{Atom}$ is called the set of *atomic propositions*. The elements of S are called *states* and $\rightarrow$ are called *the transition relation*. We write $s \rightarrow t$ for $(s, t) \in \rightarrow$.

To formalize the specification for verified systems, various modal logics and temporal logics are used. However, most of them are included in *modal μ -calculus* [16] which is a powerful logic equipped with modalities and the least fixpoints. Therefore, this paper introduces only modal μ -calculus (both of 2-valued and multi-valued). The *formulae* of modal μ -calculus are generated by the following grammar.

$$\begin{array}{lcl} \varphi ::= & P & | \neg\varphi & | \varphi \wedge \varphi \\ & \diamond\varphi & | X & | \mu X.\varphi \end{array}$$

Here, P is an atomic proposition. $\neg$ and $\wedge$ are the negation operator and the conjunction operator. So, modal μ -calculus includes the propositional logic. We write $\varphi \wedge \psi$ for $\neg(\neg\varphi \vee \neg\psi)$, $\varphi \Rightarrow \psi$ for $\neg\varphi \vee \psi$, $\Box\varphi$ for $\neg\diamond\neg\varphi$, and $\nu X.\varphi$ for $\neg\mu X.\neg\varphi[\neg X/X]$. $\diamond$ is a modal operator interpreted by using the transition relation of a Kripke structure. A state s satisfies $\diamond\varphi$ if and only if there exists a state t satisfying φ and $s \rightarrow t$. μ is called *least fixed point operator*. $\mu X.\varphi$ represents the strongest formula ψ which is equal to the result of replacing X in φ with ψ . For example, $\mu X.P \vee \diamond X$ is interpreted as follows.

$$\mu X.P \vee \diamond X = P \vee \diamond P \vee \diamond\diamond P \vee \dots$$

Therefore, a state s satisfies $\mu X.P \vee \diamond X$ if and only if there exists a path from s to a state t satisfying P .

The grammar of formulae of modal μ -calculus is subject to the side condition that $\mu X.\varphi$ has no free negative occurrences of X in φ . For example, $\mu X.\neg X$ is not a formula of modal μ -calculus. This condition is necessary to reject $\mu X.\varphi$ when there is no ψ which is equal to the result of replacing X in φ with ψ .

In conclusion, the interpretation $\llbracket \varphi \rrbracket_{K,V}: S \rightarrow \{\perp, \top\}$ of formula φ over a Kripke structure $K = (S, \rightarrow, \rho)$ and a function V is defined inductively as follows.

$$\begin{array}{ll} \llbracket \neg\varphi \rrbracket_{K,V}(s) & \stackrel{\text{def}}{=} \neg\llbracket \varphi \rrbracket_{K,V}(s) \\ \llbracket \varphi \wedge \psi \rrbracket_{K,V}(s) & \stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) \wedge \llbracket \psi \rrbracket_{K,V}(s) \\ \llbracket P \rrbracket_{K,V}(s) & \stackrel{\text{def}}{=} (s, P) \in \rho \\ \llbracket X \rrbracket_{K,V}(s) & \stackrel{\text{def}}{=} V(X)(s) \\ \llbracket \mu X.\varphi \rrbracket_{K,V} & \stackrel{\text{def}}{=} \bigwedge \{ W \mid \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \Rightarrow W \} \\ \llbracket \diamond\varphi \rrbracket_{K,V}(s) & \stackrel{\text{def}}{=} \bigvee \{ \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S, s \rightarrow t \} \end{array}$$

Here, V maps a variable X to a function $V(X): S \rightarrow \{\perp, \top\}$. For $W: S \rightarrow \{\perp, \top\}$, the function $V[X \mapsto W]$ sends X to W and the other variable Y to $V(Y)$.

φ is called *valid in K* if $\llbracket \varphi \rrbracket_{K,V}(s) = \top$ holds for any $s \in S$ and any V . For example, in the following Kripke structure K , each state (1 or 2) can reach at both of the state satisfying P and the state satisfying Q , but each state can not reach at the state satisfying both of P and Q .

$$K = (S, \rightarrow, \rho)$$

Manuscript received March 1, 2010.

Manuscript revised March 19, 2010.

[†]The author is with the Department of Information Systems, Faculty of Environmental and Information Studies, Tottori University of Environmental Studies, Tottori-shi, 689-1111 Japan.

a) E-mail: koki@kankyo-u.ac.jp

DOI: 10.1587/transinf.E93.D.2036

$$\begin{aligned}
S &= \{1, 2\} \\
\rightarrow &= \{(1, 2), (2, 1)\} \\
\rho &= \{(1, P), (2, Q)\}
\end{aligned}$$

That can be verified since $(\mu X. P \vee \Diamond X) \wedge (\mu Y. Q \vee \Diamond Y)$ is valid but $\mu X. (P \wedge Q) \vee \Diamond X$ is not.

2. Multi-Valued Model Checking

2.1 Finite De Morgan Algebra

Multi-valued model checking is studied extensively by Chechik, Gurfinkel and others [3], [11]. They give the notion of multi-valued Kripke model of temporal logics, and develop an efficient model checking algorithm with respect to such models. Their multi-valued Kripke structure takes truth values in a De Morgan algebra to represent notions related to partiality of information such as being “unknown” and “inconsistent.”

Their formulation is based on the following scenario. Given two sets X and Y , an ordinary binary relation from X to Y can be regarded as a function of $X \times Y \rightarrow 2$ where 2 is the set consisting of the truth values “true” and “false”. They replace 2 by a finite De Morgan algebra L of truth values. A *De Morgan algebra* is a structure $(L, \leq, \neg)$, where $(L, \leq)$ is a distributive lattice and $\neg: L \rightarrow L$ satisfies involution ($\neg\neg a = a$) and De Morgan laws: $\neg(a \wedge b) = \neg a \vee \neg b$, and $\neg(a \vee b) = \neg a \wedge \neg b$.

In this setting, a model is not an ordinary Kripke structure, but an L -valued Kripke structure where L is a De Morgan algebra. An L -valued Kripke structure consists of a set S of states, a function $\rightarrow$ from $S \times S$ to L , and a function ρ from $S \times \mathbf{Atom}$ to L . $\rightarrow$ is called L -valued transition relation.

A formula φ is interpreted over an L -Kripke structure $K = (S, \rightarrow, \rho)$ and a function $V: \mathbf{Var} \rightarrow (S \rightarrow L)$. The interpretation $\llbracket \varphi \rrbracket_{K,V}: S \rightarrow L$ is inductively as follows.

$$\begin{aligned}
\llbracket \neg \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \neg \llbracket \varphi \rrbracket_{K,V}(s) \\
\llbracket \varphi \wedge \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) \wedge \llbracket \psi \rrbracket_{K,V}(s) \\
\llbracket P \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \rho(s, P) \\
\llbracket X \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} V(X)(s) \\
\llbracket \mu X. \varphi \rrbracket_{K,V} &\stackrel{\text{def}}{=} \bigwedge \{ W: S \rightarrow L \mid \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \leq W \} \\
\llbracket \Diamond \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \bigvee \{ (s \rightarrow t) \wedge \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S \}
\end{aligned}$$

For example, $\{\top, \perp, m, d\}$ forms a De Morgan algebra where $\perp \leq m \leq \top$ and $\perp \leq d \leq \top$. The negation is defined as $\neg \top = \perp$, $\neg \perp = \top$, $\neg m = m$, and $\neg d = d$. m represents ‘unknown’ and d represents ‘inconsistent’. When $\rho(s, P) = \top$, $\rho(s, Q) = d$, and $\rho(s, R) = m$, the interpretation of the formula $P \vee Q$ is $\top$ and the interpretation of the formula $P \vee R$ is $\top$.

2.2 Complete Heyting Algebra

In the context of modal logics, Fitting proposes the

multi-valued logic taking truth values in a Heyting algebra that is a complete lattice [9]. His motivation is to make one multi-valued model represent a family of two-valued Kripke models on the same state set.

The author et al. apply Fitting’s multi-valued logic to system verification [13]. Multi-valued model checking performs a number of two-valued model checking all at once, by superposing many two-valued models and checking the resulting multi-valued model [26]. For example, 2^n -valued model K is regarded as the superposition of a family of two-valued models $K_1, K_2, \dots, K_n$. The validity of a modal formula ψ in K is an n -tuple of true or false. Its i th element is true if and only if ψ is valid in K_i . We extend Fitting’s modal logic to an intuitionistic variant of Kozen’s modal μ -calculus.

In fact, Boolean-valued model checking is enough for only the above motivation. Heyting-valued models are useful to consider a *multiple-experts system*, i.e., a family of two-valued Kripke models on the same state set equipped with an order $\leq$ among the models [9]. It is assumed that if $K \leq K'$ then the transition relation of K and valid atomic propositions in K are preserved in K' . Upper-closed sets of models with respect to the order form a Heyting algebra. By applying Fitting’s Heyting-valued logic to the algebra, one can get the greatest upper-closed set of models satisfying ψ as the truth value of ψ . The truth value also plays an important role in system verification. For a given model, one can consider a multiple-experts system consisting of its submodels. The truth value of ψ in the system can be regarded as an important part in the information what submodels of a given model satisfy ψ .

We take a *complete Heyting algebra* (*cHa*) (i.e., a Heyting algebra that is a complete lattice) as the structure of truth values. An L -valued Kripke structure consists of a set S , a function $\rightarrow$ from $S \times S$ to L , and a function ρ from $S \times \mathbf{Atom}$ to L .

We shall define the formulae of modal μ -calculus by extending those of intuitionistic modal logic [21] with the least and greatest fixpoints. Here, we do not follow the standard definition of modal μ -formulae, since we will be giving an intuitionistic interpretation in which $(\Diamond \varphi) \Leftrightarrow (\neg \Box \neg \varphi)$ and $(\varphi \Rightarrow \psi) \Leftrightarrow (\neg \varphi \vee \psi)$ do not necessarily hold, and we must, therefore, have $\Rightarrow$ and $\Diamond$ as primitives.

Intuitionistic modal μ -formulae are generated by the following grammar where $P \in \mathbf{Atom}$, $X \in \mathbf{Var}$, and the grammar is subject to the side condition that both of $\mu X. \varphi$ and $\nu X. \varphi$ have no free negative occurrences of X in φ .

$$\begin{array}{lcl}
\varphi ::= & P & \mid X \quad \mid \mu X. \varphi \quad \mid \nu X. \varphi \\
& \mid \Box \varphi & \mid \Diamond \varphi \quad \mid \varphi \Rightarrow \varphi \\
& \mid \perp & \mid \top \quad \mid \varphi \vee \varphi \quad \mid \varphi \wedge \varphi
\end{array}$$

We define the truth value $\llbracket \varphi \rrbracket_{K,V}: S \rightarrow L$ of an intuitionistic modal μ -formula φ inductively as follows.

$$\begin{aligned}
\llbracket \perp \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \perp \\
\llbracket \top \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \top
\end{aligned}$$

$$\begin{aligned}
\llbracket \varphi \vee \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) \vee \llbracket \psi \rrbracket_{K,V}(s) \\
\llbracket \varphi \wedge \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) \wedge \llbracket \psi \rrbracket_{K,V}(s) \\
\llbracket \varphi \Rightarrow \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) \Rightarrow \llbracket \psi \rrbracket_{K,V}(s) \\
\llbracket P \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \rho(s, P) \\
\llbracket X \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} V(X)(s) \\
\llbracket \mu X. \varphi \rrbracket_{K,V} &\stackrel{\text{def}}{=} \bigwedge \{ W \in [S, L] \mid \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \leq W \} \\
\llbracket \nu X. \varphi \rrbracket_{K,V} &\stackrel{\text{def}}{=} \bigvee \{ W \in [S, L] \mid W \leq \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \} \\
\llbracket \Box \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \bigwedge \{ (s \rightarrow t) \Rightarrow \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S \} \\
\llbracket \Diamond \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \bigvee \{ (s \rightarrow t) \wedge \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S \}
\end{aligned}$$

Operators $\Rightarrow$, $\perp$, $\top$, $\vee$, and $\wedge$ are interpreted by the corresponding structure of the complete Heyting algebra L .

For example, when $L = \wp(M)$, an L -valued Kripke structure K is regarded as a superposition of a family $(K_m)_{m \in M}$ of ordinary Kripke structures, since we have

$$m \in \llbracket \varphi \rrbracket_{K,V}(s) \iff \llbracket \varphi \rrbracket_{K_m,V}(s) = \top.$$

2.3 Min-Plus Algebra

The author et al. propose semantics of modal μ -calculus that interpret disjunctions by min and conjunctions by plus [12]. Using plus, it is possible to compute or count quantitative measures. To apply this to algebra, we adopted min-plus algebra, i.e., algebraic structures with two binary operators, min and plus. Their algebraic properties have been extensively studied and they were applied to solve problems in formal language theory, such as finite power property problem [25]. They are also widely used to analyze discrete event systems, optimization, etc. [2].

We interpret modal logic formulas on the algebra N^∞ that consists of all natural numbers and infinity ∞ . ∞ represents falsity. Finite elements represent truth, i.e., there are various levels of truth.

We shall extend the formulae of intuitionistic modal μ -calculus with the element i of N^∞ .

$$\begin{array}{lcl}
\varphi ::= & P & \mid X \quad \mid \mu X. \varphi \quad \mid \nu X. \varphi \\
& \mid \Box \varphi & \mid \Diamond \varphi \quad \mid \varphi \Rightarrow \varphi \\
& \mid \perp & \mid \top \quad \mid \varphi \vee \varphi \quad \mid \varphi \wedge \varphi \\
& \mid i
\end{array}$$

A formula φ is interpreted over an ordinary Kripke structure $K = (S, \rightarrow, \rho)$ and a function $V: \mathbf{Var} \rightarrow (S \rightarrow N^\infty)$. The interpretation $\llbracket \varphi \rrbracket_{K,V}: S \rightarrow N^\infty$ is inductively as follows.

$$\begin{aligned}
\llbracket \perp \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \infty \\
\llbracket \top \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} 0 \\
\llbracket i \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} i \\
\llbracket \varphi \vee \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \min(\llbracket \varphi \rrbracket_{K,V}(s), \llbracket \psi \rrbracket_{K,V}(s)) \\
\llbracket \varphi \wedge \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \llbracket \varphi \rrbracket_{K,V}(s) + \llbracket \psi \rrbracket_{K,V}(s)
\end{aligned}$$

$$\begin{aligned}
\llbracket \varphi \Rightarrow \psi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \begin{cases} 0 & (\llbracket \psi \rrbracket_{K,V}(s) \leq \llbracket \varphi \rrbracket_{K,V}(s)) \\ \llbracket \psi \rrbracket_{K,V}(s) - \llbracket \varphi \rrbracket_{K,V}(s) & \text{otherwise,} \end{cases} \\
\llbracket P \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \begin{cases} 0 & (s \in \rho(P)) \\ \infty & \text{otherwise,} \end{cases} \\
\llbracket X \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} V(X)(s) \\
\llbracket \mu X. \varphi \rrbracket_{K,V} &\stackrel{\text{def}}{=} \sup \{ W \in [S, L] \mid W \leq \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \} \\
\llbracket \nu X. \varphi \rrbracket_{K,V} &\stackrel{\text{def}}{=} \min \{ W \in [S, L] \mid \llbracket \varphi \rrbracket_{K,V[X \mapsto W]} \leq W \} \\
\llbracket \Box \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \sum \{ \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S, s \rightarrow t \} \\
\llbracket \Diamond \varphi \rrbracket_{K,V}(s) &\stackrel{\text{def}}{=} \min \{ \llbracket \varphi \rrbracket_{K,V}(t) \mid t \in S, s \rightarrow t \}
\end{aligned}$$

For example, the formula $\mu X. P \vee \Diamond(1 \wedge X)$ is interpreted as ∞ at s if any state where P is true is not reachable from s . If some state where P is true is reachable from s , this formula is interpreted as the length of the shortest path to a state where P is true.

3. Abstraction

While model checking is the technique to automatically verify transition systems, direct verification of large systems is often infeasible. *Abstraction* is the method to formulate and extract from concrete transition systems only that part of information necessary for verification.

Abstraction has been studied in the context of various temporal logics and modal μ -calculus. Various mathematical notions have been proposed as theoretical frameworks of abstraction: simulations [5], [10], [19], abstract interpretation [7], [20], and refinement [8], [14], [15]. An abstraction is a relation between two interpretations of the logic. We call the two interpretations, concrete interpretation and abstract interpretation. It is important that these mathematical notions satisfy the formula-preservation theorem: whenever a formula is satisfied by the abstract interpretation, it is also satisfied by the concrete one [18], [23], [24].

An abstraction helps to check that a formula is true in a concrete interpretation, called satisfaction checking. By the formula-preservation theorem, the satisfaction checking is reduced to the following subproblems:

- P1 construction of an abstract interpretation,
- P2 construction of a relation between the concrete interpretation and the abstract interpretation,
- P3 proving that the relation is an abstraction, and
- P4 checking that the abstract interpretation satisfies the formula to be verified.

It is the recent main topic to give abstractions in the multi-valued settings [11]–[13].

4. Conclusion

By applying multi-valued logics, model checking allows us to solve various problem. For example, a finite De Morgan

algebra is useful to verify ‘unknown’ systems and ‘inconsistent’ systems. A complete Heyting algebra is useful to superpose many two-valued models. A min-plus algebra is useful to solve the shortest path problem.

It is future work to introduce sound abstraction for multi-valued logics.

References

- [1] 19th Annual Symposium on Foundations of Computer Science, IEEE, Ann Arbor, Michigan, USA, Oct. 1978.
- [2] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity*, Wiley, 1992.
- [3] M. Chechik, B. Devereux, S.M. Easterbrook, A.Y.C. Lai, and V. Petrovykh, “Efficient multiple-valued model-checking using lattice representations,” *CONCUR*, pp.441–455, 2001.
- [4] E.M. Clarke and E.A. Emerson, “Design and synthesis of synchronization skeletons using branching-time temporal logic,” *Logic of Programs*, pp.52–71, 1981.
- [5] E.M. Clarke, O. Grumberg, and D.E. Long, “Model checking and abstraction,” *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp.343–354, ACM SIGACT and SIGPLAN, ACM Press, 1992.
- [6] E.M. Clarke, O. Grumberg, and D.A. Peled, *Model Checking*, The MIT Press, Cambridge, Massachusetts, 1999.
- [7] P. Cousot and R. Cousot, “Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints,” *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp.238–252, ACM Press, New York, NY, Los Angeles, California, 1977.
- [8] W. DeRoeve and K. Engelhardt, *Data Refinement: Model-Oriented Proof Methods and Their Comparison*, Cambridge University Press, New York, NY, USA, 1999.
- [9] M.C. Fitting, “Many-valued modal logics II,” *Fundamenta Informaticae*, XVII, pp.55–74, 1992.
- [10] S. Graf and H. Saidi, “Construction of abstract state graphs with pvs,” *Conference on Computer Aided Verification CAV’97*, LNCS 1254, Springer Verlag, 1997.
- [11] A. Gurfinkel and M. Chechik, “Multi-valued model checking via classical model checking,” *CONCUR*, R.M. Amadio and D. Lugiez, eds., vol.2761 of *Lect. Notes Comput. Sci.*, pp.263–277, Springer, 2003.
- [12] D. Ikarashi, Y. Tanabe, K. Nishizawa, and M. Hagiya, “Modal mu-calculus on min-plus algebra n-infty,” *Proc. Tenth Workshop on Programming and Programming Languages (PPL2008)*, Japanese Society on Software Science and Technology [1], pp.216–230, 2008.
- [13] Y. Kameyama, Y. Kinoshita, and K. Nishizawa, “Weighted Kripke structures and refinement of models,” *Proc. 23rd Conference of Japan Society for Software Science and Technology*, Tokyo, Japan, Sept. 2006.
- [14] Y. Kinoshita, P.W. O’Hearn, A.J. Power, M. Takeyama, and R.D. Tennent, “An axiomatic approach to binary logical relations with applications to data refinement,” *Lecture Notes in Computer Science*, vol.1281, pp.191–212, 1997.
- [15] Y. Kinoshita and J. Power, “A general completeness result in refinement,” *WADT’99*, LNCS 1827, Springer-Verlag, 2000.
- [16] D. Kozen, “Results on the propositional mu-calculus,” *Theoretical Computer Science*, vol.27, pp.333–354, 1983.
- [17] L. Lamport, “‘sometime’ is sometimes ‘not never’ — On the temporal logic of programs,” *POPL*, pp.174–185, 1980.
- [18] C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem, “Property preserving abstractions for the verification of concurrent systems,” *Formal Methods in System Design*, vol.6, no.1, pp.11–44, 1995.
- [19] R. Milner, “An algebraic definition of simulation between programs,” *Proc. 2nd International Joint Conference on Artificial Intelligence*, D.C. Cooper, ed., pp.481–489, William Kaufmann, London, UK, Sept. 1971.
- [20] F. Nielson, H.R. Nielson, and C. Hankin, *Principles of Program Analysis*, Springer-Verlag, 1999.
- [21] G. Plotkin and C. Stirling, “A framework for intuitionistic modal logics (extended abstract),” in *TARK: Theoretical Aspects of Reasoning about Knowledge*, 1986.
- [22] A. Pnueli, “The temporal logic of programs,” *Proc. Eighteenth Symposium on the Foundations of Computer Science*, 1977.
- [23] D.A. Schmidt, “From trace sets to modal-transition systems by stepwise abstract interpretation,” *Workshop on Structure Preserving Relations*, 2001.
- [24] K. Schneider, *Verification of Reactive Systems*, Springer-Verlag, 2003.
- [25] I. Simon, “Limited subsets of a free monoid,” *FOCS [1]*, pp.143–150, 1978.
- [26] Y. Tatsumi and Y. Kameyama, “Error detection of modeling using multi-valued model checking,” *Informal Proceedings of SIG Programming Workshop*, Information Processing Society of Japan, Japan, Jan. 2006.
- [27] M.Y. Vardi, “A temporal fixpoint calculus,” *POPL*, pp.250–259, 1988.

Koki Nishizawa received the M.S. and D.S. degrees in Information Science and Technology from The University of Tokyo in 2003 and 2006, respectively. During 2003–2008, he stayed in AIST and Tohoku University. He is a lecturer of Department of Information Systems, Faculty of Environmental and Information Studies, Tottori University of Environmental Studies, now.