# Modal Mu-calculus Extension with Description of Autonomy and Its Algebraic Structure

Susumu Yamasaki and Mariko Sasakura
*Department of Computer Science, Okayama University, Tsushima-Naka, Okayama, Japan*

Keywords: Modal Logic, Fixed Point, Abstract State Machinery, Human Computer Interaction, Application to Autonomy.

Abstract: This paper deals with complex abstract state machinery, clearly represented by modal logic with fixed point operator. The logic is well known as modal mu-calculus, which is extended to the version involving human computer interaction as well as involving awareness, communication and behavioral predicates of propositional variables as in autonomy systems. The extended version contains complexity for human machine interaction, whose meaning is represented by Heyting algebra but not by Boolean algebra. In the sense of Heyting algebra, human computer interaction of complexity can be described such that related predicates of communication and behavior may be simplified. Then the extended version can be applied to some process by means of awareness to an expertise, communication and behavior processes, and repetitions represented with fixed point operator (that is, mu-operator). This version is also concerned with model theory caused by postfix modal operator, where composition and alternation of modal operators may be organized into an algebraic structure.

## 1 INTRODUCTION

The complex information system may be more intuitively expressed if *abstract state* machinery would be taken into consideration on the assumption of the state concept:

(1) In the literatures (Droste et al., 2009; Reps et al., 2005), algebraic systems of abstract state machinery are compiled, which are related to more abstract structure of streams in the note (Rutten, 2001).

(2) On the state concept, the action causing state transitions are also significant, whether or not it is abstract or not. In the papers (Giordano et al., 2000; Hanks and McDermott, 1987), actions are captured in logical systems, while the action is a key issue in strategic reasoning.

(3) Relative to functional programs (Bertolissi et al., 2006; Thompson, 1991), the actions are to be viewed. In the book (Mosses, 1992), the procedural action is expressed by its denotation.

(4) With composed actions and programs in dynamic logic, we may see an application system of acting and sensing failures, and actions to generate and execute plans (Spalazzi and Traverso, 2000).

If we aim at the *human computer interaction* (HCI) of complexity to the programming systems,

(a) Mobile ambients (Cardelli and Gordon, 2000; Merro and Nardelli, 2005) may be effective in the sense that communication environments are well described, and

(b) From views of AI system developments, we need the bases of logics (Genesereth and Nilsson, 1987) as well as of knowledge (Reiter, 2001).

Based on the classics for beliefs and intentions, modal operations concerning mental states are captured (Dragoni et al., 1985), which are applied to form state sequences. The modal mu-calculus contains a fixed point notation to reflect some goal where actions and communications are satisfactory for given conditions. In the papers (Dam and Gurov, 2002; Kozen, 1983; Park, 1970), the proof systems with fixed point approximations are in details formulated.

In the context of AI programming system formulations, we have in this paper an extension of modal logic with fixed point operator (Venema, 2006; Venema, 2008), where modal logic with fixed point operator is settled with its transition system:

- To reflect abstract state machinery, and
- To condition the states statically containing programs and their implementations.



Yamasaki, S. and Sasakura, M.
Modal Mu-calculus Extension with Description of Autonomy and Its Algebraic Structure.
DOI: 10.5220/0009322300630071
In *Proceedings of the 5th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2020)*, pages 63-71
ISBN: 978-989-758-427-5

We here have the extended version of multi-modal mu-calculus (Yamasaki and Sasakura, 2015) for human computer interaction, with respect to algebraic meanings. It may contain a model of autonomy as recovery process, in terms of the meanings. The meaning of extended logical formulas may realize HCI such that its algebraic basis is given by Heyting algebra (corresponding to intuitionistic logic).

In this paper, it should be applied to a design of autonomy with HCI for some process, so that the designed autonomy needs the predicates of awareness (to expertise) as well as communication and behavior implementing HCI. The autonomy system can be applied to a practice for recovery from the worse condition of patients. Before making the application in details, this paper gives an outlook on the construction of the autonomy system which may provide a sequence of behavioral and communicative actions. The constructed system is initiated by awareness to the expert knowledge, by which the system specification is made. Then the system does work, as human computer interaction, with behavior (displaying such a sequence) and communication (responded by the person). The system construction is regarded as practise of the autonomy described by a formula of the modal mu-calculus extension of this paper.

The paper is organized as follows. Section 2 gives the full syntax of the modal mu-calculus extension the primary part of which we originally present. The meanings of logical formulas are defined in Section 3, where HCI may be allowable. In Section 4, the autonomy is considered from theoretical views with a practise. In Section 5, the whole logic is viewed from algebraic structures. Section 5 also deals with model theories for the logical or algebraic expression (regarded as a program), represented for (a term) forming a postfix modal operator. Section 6 concludes this paper and refers to advanced theories.

## 2 MODAL MU-CALCULUS EXTENSION

Not only for human computer interaction but also for the autonomy system components of predicates as regards awareness, communication and behavior, we have more forms extended from modal mu-calculus than our former version. That is, some predicates are newly made use of:

(i) $Aw(\varphi)$ as awareness to condition (the logical formula) $\varphi$,

(ii) $Be(\psi,\varphi,t)$ as behavior with a term $t$, for a relation between conditions $\psi$ and $\varphi$ to hold, and

(iii) $Cm(\psi,\varphi,c)$ as communication with a communication $c$, for a relation between conditions $\psi$ and $\varphi$ to hold.

The set $\Phi$ of (logical) formulas is defined inductively as follows.

$$\varphi ::= \mathrm{tt} \mid p \mid \neg\varphi \mid \sim\varphi \mid \varphi\vee\varphi \mid \mu x.\varphi \mid \langle c\rangle\varphi \mid \varphi\rangle t\rangle$$
$$\mid Aw(\varphi) \mid Be(\varphi,\varphi,t) \mid Cm(\varphi,\varphi,c)$$

Note that the intuitive meanings of symbols are described as below, where the formal meanings are given, in the next section, with the transition system as below.

(i) tt is the truth, and $p$ denotes propositions.

(ii) $\vee$ stands for the disjunction, and $\neg$ is the logical negation.

(iii) $\sim$ is another negation as interactive incapability.

(iv) $\mu$ is a least fixed point operator.

(v) $\langle c\rangle$ is a prefix modality with communication $c$.

(vi) $\rangle t\rangle$ is a postfix modality with term $t$.

(vii) $Aw$ is an awareness operator.

(viii) $Be$ is a behavior operator with respect to term $t$.

(ix) $Cm$ is a communicative operator with respect to communication $c$.

*A Transition System $\mathcal{S}$*:

For the set $\Phi$ of formulas, a transition system $\mathcal{S}$ is defined to be:

$$(S, C, U, Re, Rel, V_{pos}, V_{neg}, V_{inter}, V_{Aw}, r_{Be}, r_{Cm})$$

where:

(i) $S$ is a set of states.

(ii) $C$ is a set of labels for communications.

(iii) $U$ is a set of labels for terms.

(iv) $Re$ maps to each $c\in C$ a relation $Re(c)$ on $S$.

(v) $Rel$ maps to each $t\in U$ a relation $Rel(t)$ on $S$.

(vi) $V_{pos}, V_{neg}, V_{inter} : Prop \to 2^S$ map to each proposition (variable) a set of states, respectively.

(vii) $V_{Aw}$ is a mapping of $S$ to $2^{\Phi}$.

(viii) $r_{Be}$ is a subset of $\Phi\times\Phi$.

(ix) $r_{Cm}$ is a subset of $\Phi\times\Phi$.

# 3 MEANING OF FORMULAS FITTING HUMAN COMPUTER INTERACTION

The meaning of a formula may be a subset of the state set (in the transition system), as in Hennessy-Milner Logic (HM-Logic). However, to represent the states where human interaction may be made in computer working process, we have classified the state set into 3 subsets (parts): The first is to express the states positive to computing, the second is to designate the states for possible interaction with human, and the third is to contain the states negative to computing. In such a way, complexity is more than in HM-Logic,

Given a transition system $\mathcal{S}$, the functions

$$[\![\ ]\!]_{pos}, [\![\ ]\!]_{neg}, [\![\ ]\!]_{inter}: \Phi \to 2^S$$

are defined as meanings of formulas such that:

(i) $[\![\varphi]\!]_{pos} \cup [\![\varphi]\!]_{neg} \cup [\![\varphi]\!]_{inter} = S$, and

(ii) $[\![\varphi]\!]_{pos}$, $[\![\varphi]\!]_{neg}$ and $[\![\varphi]\!]_{inter}$ are mutually disjoint,

for $\varphi \in \Phi$.

Note that $V_{pos}(p) \cup V_{inter}(p) \cup V_{neg}(p) = S$ for each proposition (variable) $p$.

(1) $[\![\mathrm{tt}]\!]_{pos} = S$, $[\![\mathrm{tt}]\!]_{neg} = \emptyset$, and $[\![\mathrm{tt}]\!]_{inter} = \emptyset$.

(2) $[\![p]\!]_{pos} = V_{pos}(p)$, $[\![p]\!]_{neg} = V_{neg}(p)$, and $[\![p]\!]_{inter} = S \setminus ([\![p]\!]_{pos} \cup [\![p]\!]_{neg})$ $(p \in Prop)$.

(3) $[\![\neg\varphi]\!]_{pos} = [\![\varphi]\!]_{neg}$, $[\![\neg\varphi]\!]_{neg} = [\![\varphi]\!]_{pos}$, and $[\![\neg\varphi]\!]_{inter} = [\![\varphi]\!]_{inter}$.

(4) $[\![\sim\varphi]\!]_{pos} = [\![\varphi]\!]_{neg}$, $[\![\sim\varphi]\!]_{neg} = [\![\varphi]\!]_{pos} \cup [\![\varphi]\!]_{inter}$, and $[\![\sim\varphi]\!]_{inter} = \emptyset$.

(5) $[\![\varphi_1 \vee \varphi_2]\!]_{pos} = [\![\varphi_1]\!]_{pos} \cup [\![\varphi_2]\!]_{pos}$, $[\![\varphi_1 \vee \varphi_2]\!]_{neg} = [\![\varphi_1]\!]_{neg} \cap [\![\varphi_2]\!]_{neg}$, and $[\![\varphi_1 \vee \varphi_2]\!]_{inter} = S \setminus ([\![\varphi_1 \vee \varphi_2]\!]_{pos} \cup [\![\varphi_1 \vee \varphi_2]\!]_{neg})$.

(6) $[\![\langle c\rangle\varphi]\!]_{pos} = \{s \in S \mid \exists s'.\ s\ Re(c)\ s' \text{ and } s' \in [\![\varphi]\!]_{pos}\}$, $[\![\langle c\rangle\varphi]\!]_{neg} = \{s \in S \mid \forall s'.\ s\ Re(c)\ s' \text{ entails } s' \in [\![\varphi]\!]_{neg}\}$, and $[\![\langle c\rangle\varphi]\!]_{inter} = S \setminus ([\![\langle c\rangle\varphi]\!]_{pos} \cup [\![\langle c\rangle\varphi]\!]_{neg})$.

(7) $([\![\mu x.\varphi]\!]_{pos}, [\![\mu x.\varphi]\!]_{neg}) = \bigcap\{(T_{pos}, T_{neg}) \subseteq S \times S \mid ([\![\varphi]\!]_{pos\ [x:=T_{pos}]}, [\![\varphi]\!]_{neg\ [x:=T_{neg}]}) \subseteq (T_{pos}, T_{neg})\}$, and $[\![\mu x.\varphi]\!]_{inter} = S \setminus ([\![\mu x.\varphi]\!]_{pos} \cup [\![\mu x.\varphi]\!]_{neg})$, where every free occurrence of $x$ in $\varphi$ is positive such that the occurence $x$ is replaced by $T_{pos}$ and $T_{neg}$, respectively, and the operations $\bigcap$ and $\subseteq$ are componentwise.

(8) $[\![\varphi\rangle t\rangle]\!]_{pos} = \{s' \in S \mid \forall s.\ s\ Rel(t)\ s' \text{ entails } s \in [\![\varphi]\!]_{pos}\}$, $[\![\varphi\rangle t\rangle]\!]_{neg} = \{s' \in S \mid \forall s.\ s\ Rel(t)\ s' \text{ entails } s \in [\![\varphi]\!]_{neg}\}$, and $[\![\varphi\rangle t\rangle]\!]_{inter} = S \setminus ([\![\varphi\rangle t\rangle]\!]_{pos} \cup [\![\varphi\rangle t\rangle]\!]_{neg})$.

(9) $[\![Aw(\varphi)]\!]_{pos} = \{s \in S \mid \varphi \in V_{Aw}(s)\}$, $[\![Aw(\varphi)]\!]_{neg} = S \setminus [\![Aw(\varphi)]\!]_{pos}$, and $[\![Aw(\varphi)]\!]_{inter} = \emptyset$.

(10) $[\![Be(\psi,\varphi,t)]\!]_{pos} = [\![\varphi\rangle t\rangle]\!]_{pos}$ if $(\psi,\varphi) \in r_{Be}$, and $[\![\varphi]\!]_{pos}$ otherwise. $[\![Be(\psi,\varphi,t)]\!]_{neg} = [\![\varphi\rangle t\rangle)]\!]_{neg}$ if $(\psi,\varphi) \in r_{Be}$, and $[\![\varphi]\!]_{neg}$ otherwise. $[\![Be(\psi,\varphi,t)]\!]_{inter} = S \setminus ([\![Be(\psi,\varphi,t]\!]_{pos} \cup [\![(\psi,\varphi,t)]\!]_{neg})$.

(11) $[\![Cm(\psi,\varphi,c)]\!]_{pos} = [\![\langle c\rangle\varphi]\!]_{pos}$ if $(\psi,\varphi) \in r_{Cm}$, and $[\![\varphi]\!]_{pos}$ otherwise. $[\![Cm(\psi,\varphi,t)]\!]_{neg} = [\![\varphi\rangle t\rangle)]\!]_{neg}$ if $(\psi,\varphi) \in r_{Cm}$, and $[\![\varphi]\!]_{neg}$ otherseise. $[\![Cm(\psi,\varphi,t)]\!]_{inter} = S \setminus ([\![Cm(\psi,\varphi,t]\!]_{pos} \cup [\![Cm(\psi,\varphi,t)]\!]_{neg})$.

Although the definitions are clear later in terms of Heyting algebra (in Section 5), the conjunction $\wedge$ and the (Heyting algebra) implication $\longrightarrow$ are intuitively given as follows.

(i) $\varphi_1 \wedge \varphi_2$ is defined by $\neg(\neg\varphi_1 \vee \neg\varphi_2)$,

(ii) $\varphi_1 \longrightarrow \varphi_2$ is regarded as "$\varphi_1$ implies $\varphi_2$".

We have seen *term use for admissible interaction* by the definition for $[\![\text{-}]\!]_{inter}$, in terms of:

$$[\![\varphi\rangle t\rangle]\!]_{inter} = \{s' \in S \mid \exists s.\ sRel(t)s' \text{ and } s \in [\![\varphi]\!]_{inter}\}.$$

Note the usage of mu-operator for description of *term use meaning,* $\mid\ \rangle t\rangle\ \mid_{inter}$, with the fixed point operator. That is, we might have got:

$$\mid\ \rangle t\rangle\ \mid_{inter} = [\![\mu p.\ p\rangle t\rangle]\!]_{inter}.$$

# 4 AUTONOMY AND APPLICATION TO RECOVERY

With the modal mu-calculus extension, we can have a representation of an autonomy to be aware of expertise for behavior and communication with human, as well as to implement human computer interaction scheme designated by logical formulas.

### 4.1 Description of Autonomy

With awareness to an expertized condition for behaviors and communications with human (for HCI), the logical conditions might be given by means of the following formula (with paraentheses and with a propositional variable $q$ to the *mu*-operator).

$$\mu q.(((q \longrightarrow Aw(\psi)) \longrightarrow (Be(\psi,q,t) \wedge Cm(\psi,q,c))) \wedge(\vee_i(\langle c_i\rangle q \wedge q\rangle t_i\rangle)))$$

where

(1) "$\longrightarrow$" is Heyting algebra implication, whose definition is given in the next section but intuitively considered as an implication, and

(2) the terms and communications are to be implemented:

  (a) $t_i$ and $c_i$ are terms and communications within postfix and prefix modal operators, respectively, virtually in human computer interaction.

  (b) $t$ and $c$ are term (with respect to *Be*-formula) and communication (with respect to *Cm*-formula), respectively, with reference to an expertized condition by the formula $\psi$.

The mu-operator can be regarded as inductively extending the states to which conditions are kept to be satisfied, where the conditions (within the operator scope) are specified to describe awareness to expertise and repetitions of communication and behavior implementations.

### 4.2 Application to Recovery Process

As an example of recovery process which autonomy is applied to, the recovery of an aphasic patient is examined. It may be regarded as the one led by speech therapist (ST), where ST presents a picture to a patient and encourages him or her to explain the content of picture orally. Most of patients cannot explain the picture because they do hardly find proper words. We may design such a system with views of autonomy as follows, to cope with such difficulty: (i) (Expertise input) ST speaks a sound sentence at first with display of a picture, and then a patient is encouraged to repeat the sentence. (ii) (Interaction model) While they repeat ST's machinery messages and responses of the patient, the patient might possibly speak a satisfactory sentence.

With respect to the formula of the previous subsection as seen for autonomy, and the implementation need of the design by T. Kojima (2019), we may think of ST's speech as awareness of the autonomy system to ST's expertise, followed by (or implicating) behavior predicate (by interaction of ST with machinery message) and receipt of the patient with communicative predicate (including no need of receipt response), in terms of:

$$((q \longrightarrow Aw(\psi)) \longrightarrow (Be(\psi,q,t) \wedge Cm(\psi,q,c))).$$

The repeated process of messages and responses is represented, in accordance to the (sub)formula of the whole formula:

$$\vee_i(\langle c_i\rangle q \wedge q\rangle t_i\rangle).$$

The mu-operator can be seen as denoting some stable state set, with which the recovery process may supposedly continue.

*Specification of Autonomy System*:

The system (which we have implemented) provides a recovery process: The system shows pictures according to ST's interaction, with machinery messages. For instance, we can present a sentence as an exercise: "a girl eats pasta with a fork". The sentence has three nouns and one verb. Since it may be difficult for a patient to speak the whole sentence at once, we divide the sentence into three phrases of "a girl", "eats pasta", "with a fork". They are associated with pictures, as well.

The autonomy system contains three parts: "picture part", "sentence part" and "instruction part".

- In the picture part, a picture is displayed.
- In the sentence part, the sentence phrase which corresponds to a picture sequence is displayed.
- In the instruction part, the interactive instruction of ST is displayed.

As below, there is a situation in which ST interactive with a patient to speak the sentence "a girl eats pasta". The specification (which is above given) may be implemented as an autonomy system for ST by the object-oriented programming. Each part is under control of an object in a programming language, where each object has a state, such that ST may tap the screen for each object to transfer to the next state.

| Picture Part | Sentence Part |
|---|---|
| A picture of the girl | (1) "A girl"<br>(2) "eats pasta" |

| Instruction Part | |
|---|---|
| Please look at the picture. | Make a sentence. |

## 5 ALGEBRAIC STRUCTURE

We here have an algebraic structure of the domain which can be composed with respect to the meanings in the set of formulas. As a part of autonomy description, we examine the model theory of algebraic or logical expressions as a "program", which is described for the term in postfix modality.

## 5.1 Meaning and Algebra

Complexity is caused by double negation, one of which is classical and another of which is for interaction incapability. With respect to interaction capability, Heyting algebra (a bounded lattice with a specified implication: See the book (Crole, 1993) for it and category theories to semantics) is made use of, for the whole set of formulas to be represented in terms of algebraic structure.

The set $\Phi$ of formulas is related to a bounded lattice:

$$([\Phi], \bigvee, \bigwedge, [\mathrm{ff}], [\mathrm{tt}]),$$

where:

(i) $[\Phi] = \{[\varphi] \mid \varphi \in \Phi\}$ for $[\varphi] = ([\![\varphi]\!]_{pos}, [\![\varphi]\!]_{neg})$.

(ii) $\bigvee$ and $\bigwedge$ denotes a *join* and a *meet*, respectively. for which the partial order $\leq$ may be defined on the set $[\Phi]$ by means of:

$$\begin{aligned}&[\varphi_1] \leq [\varphi_2] \text{ iff}\\ &[\![\varphi_1]\!]_{pos} \subseteq [\![\varphi_2]\!]_{pos} \text{ and } [\![\varphi_2]\!]_{neg} \subseteq [\![\varphi_1]\!]_{neg}.\end{aligned}$$

(iii) $[\mathrm{ff}] = [\neg\, \mathrm{tt}] = [\sim \mathrm{tt}] = (\emptyset, S)$, and $[\mathrm{tt}] = (S, \emptyset)$.

The implication $\Rightarrow$ is equipped with, on the set $[\Phi]$:

$$[\varphi] \leq [\varphi_1] \Rightarrow [\varphi_2] \text{ iff } [\varphi_1] \bigwedge [\varphi] \leq [\varphi_2]$$

so that a Heyting algebra $[\Phi]$ is associated with the set $\Phi$ of formulas.

We then have some properties derived to denote a relation between the set $\Phi$ of formulas and the set $[\Phi]$ as model base.

**Proposition 1.** (1) $[\varphi_1 \vee \varphi_2] = [\varphi_1] \bigvee [\varphi_2]$.

(2) $[\varphi_1 \wedge \varphi_2] = [\varphi_1] \bigwedge [\varphi_2]$, where $\varphi_1 \wedge \varphi_2$ is defined as $\neg\, (\neg\, \varphi_1 \vee \neg\, \varphi_2)$.

(3) $[\sim \varphi] = [\varphi] \Rightarrow [\mathrm{ff}]$, where $[\sim \varphi] = ([\![\varphi]\!]_{neg}, S \setminus [\![\varphi]\!]_{neg})$.

*Proof.* (1)

$$\begin{aligned}&[\varphi_1 \vee \varphi_2]\\ &= ([\![\varphi_1 \vee \varphi_2]\!]_{pos}, [\![\varphi_1 \vee \varphi_2]\!]_{neg})\\ &= ([\![\varphi_1]\!]_{pos} \cup [\![\varphi_2]\!]_{pos}, [\![\varphi_1]\!]_{neg} \cap [\![\varphi_2]\!]_{neg})\\ &= [\varphi_1] \bigvee [\varphi_2].\end{aligned}$$

(2) With respect to the negation $\neg$, we can see that:

$$\begin{aligned}&[\varphi_1 \wedge \varphi_2]\\ &= ([\![\varphi_1]\!] \wedge [\![\varphi_2]\!]_{pos}, [\![\varphi_1 \wedge \varphi_2]\!]_{neg})\\ &= ([\![\varphi_1]\!]_{pos} \cap [\![\varphi_2]\!]_{pos}, [\![\varphi_1]\!]_{neg} \cup [\![\varphi_2]\!]_{neg})\\ &= [\varphi_1] \bigwedge [\varphi_2].\end{aligned}$$

(3) Since $[\varphi] \Rightarrow [\mathrm{ff}]$ is the greatest element $[\psi]$ such that $[\varphi] \bigwedge [\psi] \leq [\mathrm{ff}]$, and $[\mathrm{ff}] = (\emptyset, S)$ over the state set $S$, we have:

$$\begin{aligned}&[\sim \varphi] = ([\![\varphi]\!]_{neg}, S \setminus [\![\varphi]\!]_{pos})\\ &= [\varphi] \Rightarrow [\mathrm{ff}].\end{aligned}$$

□

We finally have a Heyting algebra implication on the set $\Phi$ of formulas, with a correspondence to the implication $\Rightarrow$ on the set $[\Phi]$.

**Definition 2.** A binary operation "$\longrightarrow$" (on the set $\Phi$) may be defined such that:

$$[\varphi_1 \longrightarrow \varphi_2] = [\varphi_1] \Rightarrow [\varphi_2].$$

## 5.2 Model of Term in Postfix Modality

To the term $t$ in the postfix modality $\rangle t \rangle$ of such a formula $\varphi \rangle t \rangle$, the relation $Rel(t)$ is assigned. In this subsection, we have a case that the term $t$ might be described by the propositional expression $F$ over a set $A_F$ ($A$, for short) of proposition letters, as below, with a correspondence to the relation $Rel(t)$ in the transition system $\mathcal{S}$:

| Syntax | Semantics |
|---|---|
| term $t$ | relation on state sets: $Rel(t)$ |
| expression $F$ | model of expression |

With reference to logical and algebraic expressions as programs, we pay attention to the propositional expression $F$ (over the set $A$ of proposition letters) is of the form:

$$\wedge_{j \in \omega}\, (l_1^j \wedge \ldots \wedge l_{n_j}^j \rightarrow l^j)$$

where $l_i^j$ denote literals, that is, propositions or their negations with *not*, and both the implication "$\rightarrow$" and the conjunction $\wedge$ are interpreted with respect to Heyting algebra. Its 3-valued models are to be examined. Note that the expression "1/2 $\rightarrow$ 1/2" is evaluated as 1. The expression $F$ as a program is descriptive, containing logical and algebraic properties.

### 5.2.1 Logic Program

The logic program with its Herbrand base is associated with the expression $F$ as above, containining the predicate *pr* (with or without "*not*" as a procedure) followed by

$$pr_1, \ldots, pr_m \text{ (as a procedural body)}$$

for $pr$, $pr_1, \ldots, pr_m$ (predicates or their negations).

This is a different view on logic programs from the one on answer set programming (Osorio et al., 2004). The model of the expression $F$ of the above form is now discussed over the 3-valued domain.

*3-Valued Model of Propositional Expression*:

For the negation "*not*" in 3-valued domain $\{0, 1/2, 1\}$ as $default\ negation$, we assume, for the proposition (letter) $p$, that:

| $p$ | $not\ p$ |
|---|---|
| 1 | 0 |
| 1/2 | 1/2 |
| 0 | 1 |

*Applicable Fixed Point as Model*:

Extending the methods of M. Fitting (1985) and A. van Gelder (1991), we may have models of least fixed points of mappings as follows, where the pair $(I, J)$ denotes the set $I$ of propositions assigned to 1, and the set $J$ of propositions assigned to 0.

For the set $A$ of proposition letters, a *monotonic* mapping is extended to this case, on the basis of the originally proposed mapping:

$$\Phi_F : 2^A \times 2^A \to 2^A \times 2^A,$$
$$\Phi_F(I_1, J_1) = (I_2, J_2),$$

is defined.

$$\begin{array}{l} I_2 = \\ \{p \mid \exists(p_1 \wedge \ldots \wedge p_i \wedge not\ p_{i+1} \wedge \ldots \wedge not\ p_j \to p) \\ \quad \text{in } F.\ \forall p_k\ (1 \le k \le i).\ p_k \in I_1, \text{ and} \\ \quad \forall p_{k'}\ (i+1 \le k' \le j)\}.\ p_{k'} \in J_1\}, \\ J_2 = \\ \{q \mid \forall(q_1 \wedge \ldots \wedge q_i \wedge not\ q_{i+1} \wedge \ldots \wedge not\ q_j \to q) \\ \quad \text{in } F.\ \exists q_k\ (1 \le k \le i).\ q_k \in J_1, \text{ or} \\ \quad \exists q_{k'}\ (i+1 \le k' \le j).\ q_{k'} \in I_1, \text{ or} \\ \quad \exists(q_1 \wedge \ldots \wedge q_i \wedge\ not\ q_{i+1} \wedge \ldots \wedge not\ q_j \to not\ q) \\ \quad \text{in } F.\ \forall q_k\ (1 \le k \le i).\ q_k \in I_1, \text{ and} \\ \quad \forall q_{k'}\ (i+1 \le k' \le j).\ q_{k'} \in J_1\}. \end{array}$$

(Extended version of the method by M. Fitting)

The least fixed point of $\Phi_F$, that is, the pair $(I, J)$ is obtained such that, with componentwise subset inclusion order $\subseteq_c$,

$$\Phi_F(I, J) = (I, J), \text{and}$$
$$\text{if } \Phi_F(I', J') = (I', J') \text{ then } (I, J) \subseteq_c (I', J').$$

The least fixed point $(I, J)$ of $\Phi_F$ may be a model of $F$, if it is consistent.
(Note: The pair $(I, J)$ is said to be *consistent*, if $I \cap J = \emptyset$.)

For the set $A$, a *monotonic* mapping is extended on the basis of the original mapping:

$$\Pi_F : 2^A \times 2^A \to 2^A \times 2^A,$$
$$\Pi_F(I_1, J_1) = (I_2, J_2),$$

is defined:

$$\begin{array}{l} I_2 = \\ \{p \mid \exists(p_1 \wedge \ldots \wedge p_i \wedge not\ p_{i+1} \wedge \ldots \wedge not\ p_j \to p) \\ \quad \text{in } F.\ \forall p_k\ (1 \le k \le i).\ p_k \in I_1, \text{ and} \\ \quad \forall p_{k'}\ (i+1 \le k' \le j).\ p_{k'} \in J_1\}, \\ J_2 = GU_F(I_1, J_1) \end{array}$$

(With $(I, J)$, $GU_F(I, J)$ is the greatest unfounded set $unfounded_F(I, J)$ inductively defined as follows).

$$\begin{array}{l} q \in unfounded_F(I, J) \\ \leftrightarrow \forall(q_1 \wedge \ldots \wedge q_i \wedge not\ q_{i+1} \wedge \ldots \wedge not\ q_j \to q) \text{ in } F. \\ \quad \exists q_k\ (1 \le k \le i).\ q_k \in J \cup unfounded_F(I, J), \text{ or} \\ \quad \exists q_{k'}\ (i+1 \le k' \le j).\ q_{k'} \in I, \text{ or} \\ \quad \exists(q_1 \wedge \ldots \wedge q_i \wedge not\ q_{i+1} \wedge \ldots \wedge not\ q_j \to not\ q) \\ \quad \text{in } F.\ \forall q_k\ (1 \le k \le i).\ q_k \in I, \text{ and} \\ \quad \forall q_{k'}\ (i+1 \le k' \le j).\ q_{k'} \in J \cup unfounded_F(I, J). \end{array}$$

(Extension of the method by A.van Gelder et al.)

The least fixed point of $\Pi_F$ is obtained as a model of $F$, if it is consistent.

### 5.2.2 Algebraic Expression

In a Heyting algebra (HA) $(A, \bigvee, \bigwedge, \perp, \top)$ equipped with the partial order $\sqsubseteq$ and an implication $\Rightarrow$:

Any expression $E$ derives some expression $F$ of the "form"

$$\bigwedge_j (x_1^j \bigwedge \cdots \bigwedge x_{n_j}^j \Rightarrow y^j),$$

where $x_i^j$ and $y^j$ are an expression $a$ or *not* $a$ (denoting $a \Rightarrow \perp$), for $a \in A$, such that

$$F \sqsubseteq E.$$

We here have a procedure to get models of a given Heyting algebra expression $F$. For the negation "*not*" in 3-valued domain $\{0, 1/2, 1\}$, we assume, for the algebraic element $a$, that:

| $a$ | $not\ a$ |
|---|---|
| 1 | 0 |
| 1/2 | 0 |
| 0 | 1 |

As a procedural way, a model construction (for the expression $F$ of the form) is presented.

*Procedure of 3-Valued Model Construction*:

(a) Assume a pair $(I,J) \in 2^A \times 2^A$ as an *input*.

(b) If the part $x_1^j \bigwedge \ldots \bigwedge x_{n_j}^j \Rightarrow y^j$ contains *not a* for $a \in J$ in the left side of the implication, then remove it from the part.

(c) If the part $x_1^j \bigwedge \ldots \bigwedge x_{n_j}^j \Rightarrow y^j$ contains *not a* for $a \notin J$ in the left side of the implication, then this part is replaced by 1 (the greatest element of $\{0, 1/2, 1\}$).

(d) Find a model $(I', J')$ for the expression obtained by repeating procedure applications of (b) and (c) (until no more procedure can be applied).

(e) If $(I,J) \subseteq (I',J')$ by pointwise (componentwise), get $(I',J')$ successfully as a *return*. Otherwise, go back to the first item (a), or halt in failure.

**Proposition 3.** *If the pair $(I',J')$ is successfully got in the Procedure of Model Construction and $I' \cap J' = \emptyset$, then it is a 3-valued model of the given expression.*

*Proof.* Observing the Procedure with a pair $(I,J)$, we may see that:

(i) By the routine (b), the expression *not a* (for $a \in J$ may be evaluated as 1, such that the part may be reduced to the (sub)expression without *not a*.

(ii) By the routine (c), the part may be evaluated as 1, such that the part may be removed from the scope of the whole meet $\bigwedge_j$.

(iii) By the routine (d), a model $(I',J')$ may be obtained without any element *not a* in the left side of any part (with an implication), because the repeated routines with (b) or (c) may remove the forms *not a*.

(iv) If $(I,J) \subseteq (I',J')$ componetwise, any part (with an implication) within the scope of the whole meet $\bigwedge_j$ may be settled as 1 by the pair $(I',J')$, since the right side of any such part contains $y^j$ of the form $b$ or *not b* (for $b \in A$).

□

# 6 CONCLUSION

With respect to abstract state machinery, human computer interaction (HCI) is included in the modal mu-calculus extension of the paper such that

(i) The meanings of formulas as conditions to the state set may be more complex, but

(ii) The meaning may be clearer on the basis of Heyting algebra.

This version has got refinements of postfix modal operator from algebraic senses. Model theories are originally constructed, in case that the postfix modal operator contains programs based on logical or algebraic expressions. A semiring structure is viewed from the point that the models of programs may cause state transitions in abstract state machine, which is given by an explicit nondeterminism description expanded from the way (Yamasaki, 2017).

Some remarks are summarized:

(i) With the version of this paper, which contains new predicates of awareness, communication and behavior, an autonomy may be designed in addition to mu-operator (least fixed point operator).

(ii) With the autonomy design, we may have a recovery system for human to practice.

(iii) A complexity of HCI is now relaxed by the description of meanings of formulas conditioning HCI, such that the description of meanings may be related to Heyting algebra.

(iv) Model theories for programming within postfix modal operator may be described on the basis of logical and algebraic methods.

As regards further refinements of this extended version of modal mu-calculus as a logical framework,

- More complex human computer interaction with some concept more cognitive, and
- More sophisticated "awareness"

should be examined.

From views of a logical framework, it should be noted that the modal mu-calculus extension of the paper contains

- the second-order propositions with fixed point operator, and
- the predicates of awareness, communication and behavior to include formulas as arguments.

As advanced works, we should have references to:

(i) Regarding the second-order (quantified) propositions, the paper (Goranko and A., 2018) treats concepts of (in)dependence functions.

(ii) The paper concerned with epistemic and intutionistic logic is to be viewed.

(iii) With respect to quantified variables by means of quantifies ranging over the set of agents, "distributed knowledge" is discussed (Naumov and Tao, 2019).

(iv) For an extension of propositional modal logic without quantification, the paper (Fitting, 2002) introduces relations and terms with scoping mechanism by lambda abstraction.

(v) Concerning the second-order predicates, the paper (Kooi, 2016) treats the concept of knowing, which is more complex than the autonomy with awareness to be designed in this paper.

(vi) As regards epistemic contradictions, the paper (Beddor and Goldstein, 2018) presents the belief predicate with the credence function of agents, which is, from the epistemic view, much more complex than the awareness predicate for autonomy system of this paper.

With respect to communication technology (Kowalski and Toni, 1996),

(i) Argumentation was, in terms of non-classical negation, formulated for lawful affairs, and

(ii) Abstract attack and defense are the argumentation concepts to have been used rather than communications for recovery processes,

while HCI may be captured in argumentation and debate theories for us to design recovery process of success and failure examinations.

From model theoretic views, it is notable that the argumentation model may be expressed by means of 3-valued logic. The 3-valued model of Heyting algebra expressions discussed in this paper is related to the semantics for defeasible reasonings able to implement argumentation (Governatori et al., 2004):

(i) Defeasibility is beforehand assumed in the given rules, to be more complex, and

(ii) The plain program consisting of rules or Heyting algebra expressions is simpler in the sense that propagation of ambiguity (caused by contradictory predicates) must be well reasoned or ruled out for its blocking.

## REFERENCES


Beddor, B. and Goldstein, S. (2018). Believing epistemic contradictions. *Review.Symb.Log.*, 11(1):87–114.

Bertolissi, C., Cirstea, H., and Kirchner, C. (2006). Expressing combinatory reduction systems derivations in the rewriting calculus. *Higher-Order.Symbolic.Comput.*, 19(4):345–376.

Cardelli, L. and Gordon, A. (2000). Mobile ambients. *Theoret.Comput.Sci.*, 240(1):177–213.

Crole, R. L. (1993). *Categories for Types*. Cambridge University Press.

Dam, M. and Gurov, D. (2002). Mu-calculus with explicit points and approximations. *J.Log.Comput.*, 12(1):119–136.

Dragoni, A., Giorgini, P., and Serafini, L. (1985). Mental states recognitiion from communication. *J.Log.Program.*, 2(4):295–312.

Droste, M., Kuich, W., and Vogler, H. (2009). *Handbook of Weighted Automata*. Springer.

Fitting, M. (2002). Modal logics between propositional and first-order. *J.Log.comput.*, 12(6):1017–1026.

Genesereth, M. and Nilsson, N. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers.

Giordano, L., Martelli, A., and Schwind, C. (2000). Ramification and causality in a modal action logic. *J.Log.Comput.*, 10(5):625–662.

Goranko, V. and A. (2018). The well-founded semantics for general logic prog kuusisto. *Review.Symb.Log.*, 11(3):470–506.

Governatori, G., Maher, M., Autoniou, G., and Billington, D. (2004). Argumentation semantics for defeasible logic. *J.Log.Comput.*, 14(5):675–702.

Hanks, S. and McDermott, D. (1987). Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33(3):379–412.

Kooi, B. (2016). The ambiguity of knowability. *Review.Symb.Log.*, 9(3):421–428.

Kowalski, R. and Toni, F. (1996). Abstract argumentation. *Artificial Intelligence and Law*, 4(3-4):275–296.

Kozen, D. (1983). Results on the propositional mu-calculus. *Theoret.Comput.Sci.*, 27(3):333–354.

Merro, M. and Nardelli, F. (2005). Behavioural theory for mobile ambients. *J.ACM.*, 52(6):961–1023.

Mosses, P. (1992). *Action Semantics*. Cambridge University Press.

Naumov, P. and Tao, J. (2019). Everyone knows that some knows: Quantifiers over epistemic agents. *Review.Symb.Log.*, 12(2):255–270.

Osorio, M., Navarro, J. A., and Arrazola, J. (2004). Applications of intuitionistic logic in answer set programming. *TLP*, 4(3):325–354.

Park, D. M. R. (1970). Fixpoint induction and proof of program semantics. *Machine Intelligence*, 5:59–78.

Reiter, R. (2001). *Knowledge in Action*. MIT Press.

Reps, T., Schwoon, S., and Somesh, J. (2005). Weighted pushdown systems and their application to interprocedural data flow analysis. *Sci.Comput.Program.*, 58(1-2):206–263.

Rutten, J. (2001). *On Streams and Coinduction*. CWI.

Spalazzi, L. and Traverso, P. (2000). A dynamic logic for acting, sensing and planning. *J.Log.Comput.*, 10(6):787–821.

Thompson, S. (1991). *Type Theory and Functional Programming*. Addison-Wesley, Amsterdam.

Venema, Y. (2006). Automata and fixed point logic: A coalgebraic perspective. *Inf.Comput.*, 204(4):637–678.

Venema, Y. (2008). *Lectures on the Modal Mu-Calculus*. ILLC, Amsterdam.

Yamasaki, S. (2017). Semantics and algebra for action logic monitoring state transition. In *Proceedings of 2nd International Conference on Complexity, Future Information Systems and Risk, Complexis*, pages 110–115.

Yamasaki, S. and Sasakura, M. (2015). Multi-modal mu-calculus semantics for knowledge construction. In *Proceedings of 7th IC3K, KEOD*, pages 358–363.