Multi-modal Mu-calculus Semantics for Knowledge Construction

Susumu Yamasaki[1] and Mariko Sasakura[2]
[1]*HCI Group, Okayama University, Okayama, Japan*
[2]*Department of Computer Science, Okayama University, Okayama, Japan*

Keywords: Multi-modal Logic Semantics, Action Logic, Knowledge Construction.

Abstract: This position paper aims at setting a new semantics for multi-modal mu-calculus to represent interactive states where abstract actions may be applied to. A least fixed point formula may be available to denote states allowing interaction. A simple algebraic representation for interactive states can be definable. For communication between human and machinery, a modality is reserved. In applicative task domains, knowledge construction is focused on with respect to interactive action applications through communications. Panel touch behaviour on iDevice as practice, URL references as functions and grammatical rule applications for sequential effects are studied, as knowledge construction technologies. These views coherent with abstract state machine are finally related to recent trends as semiring in algebraic structure and coalgebra for streams as sequential knowledge structures. A refinement of interactive techniques is positioned into a formal approach to multi-modal logic, applicable to some practices.

1 INTRODUCTION

This positioning is motivated by an intention to present the unified machinery framework of *action* in knowledge construction and *interactive* communication with human ideas, for a human machine interaction as illustrated below, where the environments of human and machinery are virtually regarded as states.

Human		Machinery
Communications	$\Rightarrow$	Reasoning
Cognition	$\Leftarrow$	Actions

As actions of both artistic and technological methods with respect to knowledge engineering in interactive artificial intelligence, this positioning supposes working (action as reasoning) in (i) *design* of paper folding to make some forms, (ii) *knowledge acquisition* by references to URLs, and (iii) *grammatical rule application* for language learnings, as case studies.

As the book (Jackson, 1989) describes, the art of paper folding is rich enough in terms of simple and beautiful fascinations. Anyone can do anywhere, anytime by means of papers which are also attractive in practices as well as fine displays. With respect to interactive computing and convenience, iDevice panel touch, as action, may be interesting for the art. Compared with the paper as a medium, 2D panel touch is simpler even for knowledge construction to the 3D form made by paper medium. However, simplicity of panel touch may cause difficulty in graphical visualizations. This is regarded as trade off for simplicity and compactness automated by modeling and mechanized panel touch. This positioning aims at design for implementation methods and tools as reasoning aspects, in respect to machanized action and interaction with human.

As regards URLs, it may involve knowledge construction by means of location references such that acquisition of knowledge can be implemented as actions to have insights into contexts. Observing and enjoying knowledge construction can be interactive to human behaviours with automated eInfrastructure.

Concerning language learning, grammatical rule applications are respected as in case of recovery from language incapability written in the book (Chapey, 2001). The cognitive process of clients often needs interactions to other human, whose work may be partially realized by machine intelligences. To recover language capability or to learn more, the cognitive process must be supported with respect to and consistently by formally grammatical rules.

For a method of unifying machinery with actions and interaction to human communication, we refine multi-modal logic as representation of moni-

Yamasaki, S. and Sasakura, M..
Multi-modal Mu-calculus Semantics for Knowledge Construction.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 358-363
ISBN: 978-989-758-158-8

toring *states* (implementable environments). Semantics is newly refined with ideas from Hennesy-Milner Logic denotation to states and Keene-Kripke model for modality as well as fixed point theory. By means of states, interaction can be made and programmed actions are implemented. This is a primary purpose of this positioning. As another side, state-constraint mechanism is viewed from the concept like *abstract state machine* originating from the works of Gurevich,Y. As refined techniques, we classify an action into a postfix-modality following a formula, where interactions are captured with communication modality (of prefix-modality preceding a formula). Thus interaction is acknowledged even in the formulation of logic. We then consider a sequence of actions. It is related to the sequential process modeled in modal logic. It is also relevant to *semiring* structure with respect to state-transitions in the handbook and papers (Droste et al., 2009; Reps et al., 2005).

The positioning is organized as follows. Section 2 is to formulate multi-modal mu-calculus with semantics regarding interaction for monitoring the state sets. It involves an interactive state-constraint implementation for actions. A fixed point formula may be made use of in representations of actions at interactive states. Section 3 presents some working in progress, practical or theoretical. This section contains (i) realization of folding paper by iDevice, as a state-constraint implementation, (ii) acquisition through URL references, as functions, and (iii) grammatical rule sequence with state-constraints. Section 4 gives some remarks regarding formal model of this positioning.

2 MULTI-MODAL MU-CALCULUS FOR STATE-CONSTRAINT

We make use of actions (Mosses, 1992), Hennessy-Milner logic with reference to the papers (Cardelli and Gordon, 2000; Merro and Nardelli, 2005; Hennessy and Milner, 1985; Milner, 1999) and multi-modal μ-calculus for modeling of state constraint implementation. With reference to multi-process monitoring states regarding negations "$\sim$ vs. $\neg$", we have the set Φ of formulas defined by:

$$\varphi ::= \mathrm{tt} \mid p \mid \neg\varphi \mid \sim\varphi \mid \varphi \vee \varphi \mid \langle c\rangle\varphi \mid \mu x.\varphi \mid \varphi\rangle a\rangle$$

We here have a (standard) prefix modality $\langle c\rangle$ (for communication), a postfix one $\rangle a\rangle$ (for action), and a negation (sign) $\sim$ as regards incapability of interaction, in addition to standard propositions p, the logical negation $\neg$ and a least fixed point operator μ.

The semantics for formulas are definable on the basis of *a transition system*, which is modified and extended for denoting "interaction" states where some implementation and action are available. The transition system is

$$\mathcal{S} = (S, C, Ac, Re, Rel, V_{pos}, V_{neg}, V_{inter}),$$

where:

(i) S is a set of states.

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

(iii) Ac is a set of actions.

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

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

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

The reason why 3 assignments of V_{pos}, V_{neg} and V_{inter} are adopted comes from a motivation to introduce an assignment for monitoring *interaction* and the existence of negation "$\neg$". Given a transition system $\mathcal{S}$, the functions $[\![\ ]\!]_{pos}, [\![\ ]\!]_{neg}, [\![\ ]\!]_{inter}$: $\Phi \to 2^S$ are defined 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 $\varphi \in \Phi$, while $[\![\sim\varphi]\!]_{inter} = \emptyset$,

to demonstrate that the formula (process) denotation, the state set for interaction, is empty.

Meaning concerned with two modalities $\langle c\rangle$, $\rangle a\rangle$:

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

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

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

(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) $[\![\langle c\rangle\varphi]\!]_{pos} = \{s \in S \mid \exists s'.\ s\ Re(c)\ s' \wedge s' \in [\![\varphi]\!]_{pos}\}$, $[\![\langle c\rangle\varphi]\!]_{neg} = \{s \in S \mid \forall s'.\ s\ Re(c)\ s' \Rightarrow s' \in [\![\varphi]\!]_{neg}\}$, 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 φ is positive.

(8) $[\![\varphi\rangle a\rangle]\!]_{pos}$
$= \{s' \in S \mid \forall s.\ s\, Rel(a)s' \Rightarrow s \in [\![\varphi]\!]_{pos}\}$,
$[\![\varphi\rangle a\rangle]\!]_{neg}$
$= \{s' \in S \mid \forall s.\ s\, Rel(a)\ s' \Rightarrow s \in [\![\varphi]\!]_{neg}\}$,
$[\![\varphi\rangle a\rangle)]\!]_{inter} = S \setminus ([\![\varphi\rangle a\rangle]\!]_{pos} \cup [\![\varphi\rangle a\rangle]\!]_{neg})$.

(Note) Implementation sense of modality $\langle c\rangle$ is from communication labelled by c, like the standard modality. Modality $\rangle a\rangle$ possibly comes from actions which machinery virtually causes. When it is applied to a state s, it conceives a relation $Rel(a)$.

By the definition for $[\![\text{-}]\!]_{inter}$ to be concerned with denoting admissible interaction, we can see that:

$$[\![\varphi\rangle a\rangle]\!]_{inter} = \{s' \in S \mid \exists s.\ sRel(a)s' \wedge s \in [\![\varphi]\!]_{inter}\}.$$

Some *algebraic* treatments are available with respect to denoting interactive states.

(A1) We have Heyting algebra

$$\mathcal{H} = (\{0, 1/2, 1\}, \leq, \bigvee, \bigwedge, 0, 1, \longrightarrow),$$

where $0 \leq 1/2 \leq 1$, and the expression $a \longrightarrow b$ denotes a greatest element c of $\{0, 1/2, 1\}$ such that $a \bigwedge c \leq b$. We have a semantic function *Mon*, to see whether a process (supported by a formula) on a state is legal (by value 1/2) for interaction:

$$Mon : \Phi \rightarrow S \rightarrow \{0, 1/2, 1\},$$
$$Mon[\![\varphi]\!]s = \begin{cases} 1 & s \in [\![\varphi]\!]_{pos} \\ 1/2 & s \in [\![\varphi]\!]_{inter} \\ 0 & s \in [\![\varphi]\!]_{neg} \end{cases}$$

Given a transition system $\mathcal{S}$ and $\mathcal{H}$ for monitoring, with $\varphi \in \Phi$ and $s \in S$:

(i) $Mon[\![\neg\varphi]\!]s =$ if $s \in [\![\varphi]\!]_{inter}$ then $Mon[\![\varphi]\!]s$ else $Mon[\![\varphi]\!]s \longrightarrow 0$.

(ii) $Mon[\![\sim\varphi]\!]s = Mon[\![\varphi]\!]s \longrightarrow 0$.

(iii) $Mon[\![\sim\neg\varphi]\!]s \leq Mon[\![\neg\sim\varphi]\!]s$.

(iv) $Mon[\![\varphi_1 \vee \varphi_2]\!]s = Mon[\![\varphi_1]\!]s \bigvee Mon[\![\varphi_2]\!]s$.

(v) If $s\ Re(c)\ s'$, and $Mon[\![\langle c\rangle\varphi]\!]s = 1/2$, then $Mon[\![\varphi]\!]s' = 1/2$.

(vi) If $s\ Rel(a)\ s'$, and $Mon[\![\varphi]\!]s = 1/2$, then $Mon[\![\varphi\rangle a\rangle]\!]\ s' = 1/2$.

(A2) A fixed point formula is applicable to the modality denotations. The meaning of formulas $\varphi\rangle a\rangle$ contains the states, to which the states supported by the formula (process) φ might transit. With the fixed point formula,

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

may be regarded as meanings with the functions a within modalities, respectively, for *interaction*.

3 KNOWLEDGE CONSTRUCTION

The formula, say φ, is considered as a process (governing the states), abstracted from an interaction scheme and cognition as follows.

As suggested later, the panel touch in iDevice is represented by action in modality $\rangle a\rangle$. As regards the cognition of concepts with *references*, technologies of the internet URLs are available such that modality $\rangle a\rangle$ can be applied. As to learning grammatical *rules*, modality $\rangle a\rangle$ may be adopted. The modalities are to be conveniently placed as followers, because they are concerned with the roles (effects) of actions in formal reasoning:

Interaction Scheme
Interaction with communication (C) and action (A) between human (H) and machinery (M) as artificial intelligence:

H	$\overset{C}{\rightarrow}$	M	: Communications
$\langle c\rangle\varphi$		φ	: Supporting formulas
M	$\overset{A}{\rightarrow}$	H	: Actions
φ		$\varphi\rangle a\rangle$	: Presentations of actions

Cognition
Human (H) cognition of action (A):

H	$\overset{A}{\rightarrow}$	H' (advanced H)	: Cognition
φ		$\varphi\rangle a\rangle$	: Acquiring actions

3.1 Interactive Paper Folding

Folding Model
We assume some points for an art of folding paper (*origami*) to be virtually mechanized or implementable by iDevice, while folding is an action in modal operator at states:

- An origami contains a set of faces.
- A face is an area of no thicknesss, enclosed with edges. A face is, for iDevice techniques, restricted to a triangle of 3 edges and 3 vertexes, while the initial sheet paper is supposedly regarded as containing 2 triangles.
- A crease line is an edge adjacent to 2 faces.
- An edge is a line segment ended by 2 vertexes.

Making origami is (i) to specify a crease line, and (ii) to fold 2 adjacent faces with an angle between 0 and 180 degrees.

Valley and Mountain Foldings
A primitive but so fundamental folding is to specify a crease line, folding 2 faces like a valley or a mountain, called *valley folding* (V-fold) or *mountain folding* (M-fold), respectively.

As implementations of V-fold and M-fold on iDevices (virtually with correspondence to communication and action modal operators), we have 2 alternative methods below. Each of them is interpreted as an abstract action constrained by a state modeling the modal logic formula in the previous section, and as abstractly causing a transition to the next state after V-fold or M-fold virtually results.

(I1) It is to determine the positions of 2 vertexes: A panel touch with *long-press* from the first point and with *pan* (drag) to another suggests a line segment. For the suggestive line segment, a perpendicular is provided as an effective crease line, at already-decided position crossing the given (line segment), so that V-fold or M-fold might be implemented.

(I2) It is to directly by touch specify a crease line: By an operation *pinch in/out*, a crease line is provided between the suggested points so that *flick* operation might be effective for either face concerning the crease line to be folded.

They are basic tools for a more refined visualisation (Sasakura et al., 2013).

Sequential Foldings
As a sequence formation for making (flying) *plane*, we can now have a sequence of folding by recursion (cycle):

(i) At a state of *communicating* (C), machinery sees by interaction where machinery makes folding and how it does, and transit to the next concerning state: This is regarded as monitored by a formula of the form $\langle c\rangle\varphi$ with interaction admission, in which communication c virtually represents an interaction.

(ii) At the next state of *reasoning* (R), machinery makes an implementation of iDevice for folding determined in the previous step on the point and by the method: This is regarded as monitored by a formula of the form φ with legal interaction admission.

(iii) We then see that a *folding* (F) is made by the previous implementation: This is regarded as monitored by a formula of the form $\varphi\rangle a\rangle$ with interaction admission.

The cycle is regarded as monitoring realized by formula denotations in the following manner, where we take abbreviations for C, R and F:

Monitoring:	$\langle c\rangle\varphi$	φ	$\varphi\rangle a\rangle$
	|	|	|
Interaction for:	C	R	F

In a more concretized folding to an airplane, a whole sequence, virtually causing state-transitions for implementation, is given with an interactive communication at each state (step):

(i) V-fold, to make the rightmost and uppermost position set into the centre.

(ii) V-fold, to make the leftmost and uppermost position set into the centre.

(iii) V-fold, to operate by a central and vertical crease line.

(iv) V-fold, to operate for one out of 2 faces (made in step (iii)) by a crease line parallel to the crease line of step (iii), with an angle of 90 degrees open to the outer side.

(v) V-fold, to operate for another out of 2 faces (made in step (iii)) by a crease line parallel to the crease line of step (iii), with an angle of 90 degrees open to the outer side.

A visualization of a simple plane may be designed, on the basis of the above action sequence.

3.2 Reference Recursion

As URL structures, a referenced page (named x) may contain some senses recursively linked with other references, as a function $x \mapsto (y_1,\ldots,y_n)$ $(n \geq 0)$ with references x, and $y_1, \ldots, y_n$.

Let X be a set of references. With respect to modality $\rangle a\rangle$, $u \equiv u_a$ is definable as a function, $u : X \to 2^X$. The function $\hat{u} : 2^X \to 2^X$ is defined for the function u by: $\hat{u}(Y) = \cup_{x\in Y}\, u(x)$. Note for any $\hat{u}$ that $\hat{u}(\emptyset) = \emptyset$. For the function $1_X : X \to 2^X$, $1_X(x) = \{x\}$, the function $\hat{1}_X$ is $\hat{1}_X(Y) = Y$, the identity on 2^X. Let U_X be a set of functions of 2^X to 2^X such that $f(\emptyset) = \emptyset$ for $f \in U_X$. The composition of f and g in U_X, $g \circ f : 2^X \to 2^X$ is $(g \circ f)(Y) = g(f(Y))$. Then, the composition is associative. With respect to the identity element $\hat{1}_X$, $f \circ \hat{1}_X = \hat{1}_X \circ f = f$.

Then $\langle U_X, \circ, \hat{1}_X\rangle$ is a *monoid* (semigroup with identity).

On the other hand to the operation $\circ$, the alternation $+$ is considerable. $f + g : 2^X \to 2^X$ is defined to be $(f+g)(X) = f(X) \cup g(X)$. It is seen that the operation $+$ is commutative and associative. With the function $0_X : X \to 2^X$ such that $0_X(x) = \emptyset$, $\hat{0}_X(Y) = \emptyset$.

Then, as the identity element $\hat{0}_X$, $f+\hat{0}_X = \hat{0}_X + f = f$. It is clear from the definition of "+" that $f+f=f$ holds (*idempotence*).

As above mentioned, $\langle U_X, +, \hat{0}_X \rangle$ is a *commutative* monoid. As regards the composition, we can see for any $f \in U_X$ that $\hat{0}_X \circ f = f \circ \hat{0}_X = \hat{0}_X$, because $\hat{0}_X(Y) = \emptyset$. By properties of the operations $+$ and $\circ$, on the set U_X, we have:

Proposition 1. $\langle U_X, +, \circ, \hat{0}_X, \hat{1}_X \rangle$ is an (idempotent) semiring.

Proof. It remains to see distributive laws, which hold with the following reasons:

$$\begin{aligned}(g+h)\circ f(Y) &= g(f(Y)) \cup h(f(Y)) \\ &= (g\circ f + h\circ f)(Y) \\ f\circ(g+h)(Y) &= f(g(Y)\cup h(Y)) \\ &= (f\circ g + f\circ h)(Y)\end{aligned}$$

Q.E.D.

We now examine the property of compositional sequences of functions of the form $\hat{u}$.

Definition 2. The composition of $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{n-1}$ and $\hat{u}_n$ is successful for $x \in X$ if $\hat{u}_n \circ \ldots \circ \hat{u}_1(\{x\}) = \emptyset$.

Let U_X^h ($\subseteq U_X$) be a set of functions of the form $\hat{u}$, $\hat{1}_X$ or $\hat{0}_X$. To provide a composition (sequence) σ of the functions from U_X^h, successful for a given $x \in X$, we have a recursive procedure $Pro(x)$ as follows, where $\perp$ stands for a failure: $\perp + \sigma = \sigma + \perp = \sigma$ and $\perp \circ \sigma = \sigma \circ \perp = \perp$ for any finite sequence σ constructed by the functions from U_X^h.

Procedure *Pro*:

$Pro(x, U_X^h) \Leftarrow$ if $U_X^h = \emptyset$ then $\perp$
else $+_{\{\hat{u} \in U_X^h\}} Check(x,u)$

$Check(x,u) \Leftarrow$ if $u(x) = \emptyset$ then $\hat{u}$
else $\circ_{\{y \in u(x)\}}\ Pro(y, U_X^h - \{\hat{u}\}) \circ \hat{u}$

Proposition 3. On the basis of the procedure *Pro* for a given $x \in X$, $Pro(x)$ contains a non-$\perp$ sequence iff some sequence from $Pro(x)$ is successful for x.

Proof. (1) If $Pro(x, U_X^h)$ contains a sequence successful for x, then it must be a non-$\perp$ sequence, by the construction of the procedure *Pro* with *Check*.
(2) Assume that $Pro(x, U_X^h)$ contains a non-$\perp$ sequence. Induction is made on recursion included in *Pro* with *Check*: (i) If some u exists such that $u(x) = \emptyset$, then $Pro(x, U_X^h)$ contains $\hat{u}$, successful for x.
(ii) If $u(x) \neq \emptyset$, suppose each procedure $Pro(y, U_X^h - \{\hat{u}\})$ for $y \in u(x)$, with the preceding function $\hat{u}$, in $Check(x,u)$. By the procedure $Pro(y, U_X^h - \{\hat{u}\})$ to (by induction hypothesis) contain a sequence successful for y, and by distributive laws of $\circ$ over +, there may be a sequence from U_X^h, beginning with $\hat{u}$ (as in $Check(x,u)$), successful for the given x. This concludes the induction step. Q.E.D.

3.3 Rewriting Rules

Learning the rules r and r', the human's state may be transited from s to s', which can be mechanized in *rewritings with states*:

$$\begin{array}{cccc} \text{state} & & \text{state} & \\ s & \xrightarrow{r} & s' & \xrightarrow{r'} \end{array}$$

To intuitively see such a structure, we now have a function sequence virtually with reference to states. Let Nt and Σ be a set of nonterminals and a set of terminals, respectively. With respect to modality $\rangle a \rangle$, a rule r regarded as an action a is defined to be a function $r : Nt \cup \Sigma \to (Nt \cup \Sigma)^*$ such that $(Nt \cup \Sigma)^*$ is the set of all finite sequences, formed from the set $Nt \cup \Sigma$, containing the nil sequence nil, and r(t) = t for any $t \in \Sigma$. The function r can be extended to the one $\overline{r} : (Nt \cup \Sigma)^* \to (Nt \cup \Sigma)^*$ as defined to be

$$\begin{aligned}&\overline{r}(nil) = nil, \text{ and} \\ &\overline{r}(z) = cons(r(head(z)), \overline{r}(tail(z))),\end{aligned}$$

where (i) *head* takes the first symbol from a given non-*nil* sequence, (ii) *tail* is a sequence constructed by cutting off the first symbol for a given sequence, and (iii) *cons* is an operation to get a sequence by combining a symbol with a sequence.

The composition of $\overline{r}_1$ and $\overline{r}_2$ can be defined to be

$$(\overline{r}_2 \bullet \overline{r}_1)(z) = \overline{r}_2(\overline{r}_1(z))$$

with the identity function $\overline{1}(z) = z$ for any $z \in (Nt \cup \Sigma)^*$. As regards the composition, the associative laws holds. Now let

$V_\Sigma = \{fn \mid fn$ is a function of $(Nt \cup \Sigma)^* \to (Nt \cup \Sigma)^*\}$.

Then $\langle V_\Sigma, \bullet, \overline{1} \rangle$ is a monoid.

Given a nonterminal $m \in Nt$, whether there is a sequence $\overline{r}_1, \ldots, \overline{r}_n$ such that $(\overline{r}_n \bullet \ldots \bullet \overline{r}_1)(m) \in \Sigma^*$ can be decided, if the set Nt is finite: Neglecting the elements of Σ, at least a similar procedure like the procedure *Pro* (as regards reference completion for successful termination) can work for a given nonterminal.

4 CONCLUDING REMARKS

We have summaries regarding this work in progress for formal methods in interactive techniques.

(i) This formality of multi-modal logic, to monitor states virtually interactive for iDevice may be closely relevant to the recent trend of game semantics (Venema, 2008).
(ii) As well as newly presented semantics for this calculus based on a state set, actions regarding a modal

operator are discussed, with respect to their functional and algebraic aspects and with reference to works (Giordano et al., 2000; van der Hoek et al., 2005; Kucera and Esparza, 2003). The accounts of actions are made by applications of Heyting algebra, fixed point theory, and *semiring* structure. They are also related to trend *coalgebra* (Kurz, 1989).
(iii) Fixed point logic (Venema, 2006) may include the present version, since the action modality may be denoted by a fixed point. However, the mu-operator requires some restriction that the operator may be associated with a monotone function. For a nonmonotonic case, we have backgrounds (Genesereth and Nilsson, 1987; Yamasaki, 2006; Yamasaki, 2010).
(iv) As regards sequence formation in iDevice, it is closely related to knowledge structure, where the well-done sequence presents a beauty based on mechanized formation of reasonable (simple) state-transitions. Whether well mechanized formations of a sequence for the origami *crane* by iDevice would be a problem from the views of interaction techniques with graphical designs of practical impacts. Concerning URL references, a referential closeness is discussed with respect to the idea of successful sequence of references. The sequence is related to the structure of semiring, captured by coalgebraic behaviours. For a task as implementing heuristic cognition stages of recovery from language, it is a methodological or technical idea to automate grammatical rule applications, while the discussion of this positioning is really concerned with state-constraint grammars between context-free and context-sensitive grammar hierarchies (Kasai, 1970), simpler than the class of constraint functional programming (Bertolissi et al., 2006) and more classical than the recent studies of (infinite) streams and languages by coalgebra (Rutten, 2001; Winter et al., 2013; Winter et al., 2015). We might, however, have formal reasonings to implementing cognitions or to making them graded up as artificial (machine) intelligences, following the advanced theories from those (Genesereth and Nilsson, 1987; Reiter, 2001).

REFERENCES

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.

Chapey, R. (2001). *Language Intervention Strategies in Aphasia and Related Neurogenic Communication Disorder*. Lippincott Williams & Wilkins.

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

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.

Hennessy, M. and Milner, R. (1985). Algebraic laws for nondeterminism and concurrency. *J.ACM*, 32(1):137–161.

Jackson, P. (1989). *The Complete Origami Course*. Gallery Books.

Kasai, T. (1970). An hierarchy between context-free and context-sensitive languages. *J.Comput.Syst.Sci.*, 4(5):492–508.

Kucera, A. and Esparza, J. (2003). A logical viewpoint on process-algebraic quotients. *J.Log.Comput.*, 13(6):863–880.

Kurz, A. (1989). *Coalgebra and Modal Logic*. CWI, Amsterdam.

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

Milner, R. (1999). *Communicating and Mobile Systems: The Pi-Calculus*. Cambridge University Press.

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

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, Amsterdam.

Sasakura, M., Tanaka, K., Yamashita, E., Tanabe, H., and Kawakami, T. (2013). A bi-phase model of folding origami interactively with gap representation. In *17th International Conference on Information Visualisation: Proceedings*, pages 494–498.

van der Hoek, W., Roberts, M., and Wooldridge, M. (2005). A logic for strategic reasoning. In *4th AAMAS: Proceedings*, pages 157–164.

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.

Winter, J., Marcello, B., Bonsangue, M., and Rutten, J. (2013). Coalgebraic characterizations of context-free languages. *Formal Methods in Computer Science*, 9(3):1–39.

Winter, J., Marcello, B., Bonsangue, M., and Rutten, J. (2015). Cntext-free coalgebra. *J.Comput.Syst.Sci.*, 81(5):911–939.

Yamasaki, S. (2006). Logic programming with default, weak and strict negations. *Theory.Pract.Log.Program.*, 6(6):737–749.

Yamasaki, S. (2010). A construction of logic-constrained functions with respect to awareness. In *LRBA, MALLOW: Proceedings*, pages 84–98.