Semantics and Algebra for Action Logic Monitoring State Transitions

Susumu Yamasaki
HCI Group, Okayama University, Okayama, Japan

Keywords: State Constraint Agent, Interactive Monitoring, Modal Logic and Algebra.

Abstract: This position paper is concerned with aspects of an interactive state transition system (based on abstract state machine) by means of the state monitoring in action logic (as multi-modal logic), towards a step to the design for complex systems. Logical models are here presented as theories for implementation design on iDevice, with respect to the algebraic structure caused by state transitions. As a simpler design of complex AI, the environmental constraint is captured as a state, where the function applications are available at each state with the transition to the next states. For communication to the state, and function applications at the state, multi-modal logic model may be of use, where the formula or the condition monitors the state. Then interaction availability is significant, expressed in some algebra on the basis of the meaning definitions for formulas (conditions). By the state transition system, URL searching operations are now formally considered as in algebraic structure. The application of predicates to the states is regarded as applications of functions (transforming conditions) such that its algebraic structure may be given.

1 INTRODUCTION

As methods of monitorng and analysis for systems, this positioning is concerned with the following concepts, for extensions of action logic (as multi-modal logic): (i) interaction modeling for abstract state machine as a state transition system, (ii) monitoring states in action logic with modal operators, (iii) logic of action, with respect to multi-modal logic and applications of functions, and (iv) denotational semantics for action logic.

Abstract state machine (by Y. Gurevich) can be a basis for the framework of state-transition systems applicable to complex AI systems. Containing state-transitions, action logic is needed for design methods even on iDevice, as in the paper (Yamasaki and Sasakura, 2015).

For the required design methods, interactive stage would be here formulated as being monitored, to reflect behaviours on popular iDevices (for interactive AI-tools), with relevance to usage of ideas in functional programming (Thompson, 1991) or process algebra (Cardelli and Gordon, 2000; Milner, 1999) to AI-tools. Combined with abstract state machine, the function applications can be made with state transitions. However, the state can be nowadays realized by a panel display to be touched on iDevices. The iDevice is, on one hand, to present the state transition where the function may be applied. On the other hand, it is interpreted to support an interactive process, in which function applications are executed in a programming system so that the evaluation may be obtained.

The state transition system is involved in action logic (Hennessy and Milner, 1985; Kucera and Esparza, 2003) and modal logic (Venema, 2008). Based on the new aspect of the meaning definition for formulas (or conditions), an interaction state (where interactive actions like communications and/or function applications are available) can be presented.

As regards knowledge-based systems, URL searching forward and backward would be examined with respect to the operations (composition – multiplication– and alternation – addition –) of function applications in a state transition system. The view on the operations is closely related to automata theory as in the book (Droste et al., 2009) where algebraic structures like semiring are compiled. In this positioning, a new technique on the reduction of multiplicative inverse is to be presented, after the URL searching is well organized as an algebraic structure, caused by a state-constraint system. For knowledge-based systems (which involve technologies in state transition systems as well as in action logic), actions by predicates or logical formulas may be regarded as function applications.

The position paper is organized as follows. In Section 2 we observe Yale Shooting Problem as an intro-

Yamasaki, S.
Semantics and Algebra for Action Logic Monitoring State Transitions.
DOI: 10.5220/0006345901100115
In *Proceedings of the 2nd International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2017)*, pages 110-115
ISBN: 978-989-758-244-8

ductory motivation to this positioning. It is in Section 3 followed by a revised formulation of multi-modal mu-calculus to monitor the state transition system. In Sections 4 and 5, some function applications of state transition systems are discussed from algebraic views. A concluding remark, with related topics, is briefly given, as well.

2 A KNOWLEDGE-BASED SYSTEM FOR ACTIONS

Solution Display of Yale Shooting Problem

The solution of the AI problem is often expressed by a sequence of state-transitions. The state is temporally prepared for and the operation is equipped with, where the formula or condition is attached to the state. AI solution and display can thus be made in terms of (interactive) state-constraint system. Let us have an outlook on the display of solutions in the Yale Shooting Problem (Hanks and McDermott, 1987).

The *Yale Shooting Problem* is a scenario in logic:

(i) A turkey is initially alive and a gun is initially unloaded.

(ii) Loading the gun, the shooter waits for a moment.

(iii) Then shooting at the turkey is expected to kill it.

The scenario is captured with the condition changing truth values over time, like *alive* and *loaded*. Assuming four time points 0, 1, 2, and 3, $alive(t)$ and $loaded(t)$ supposedly denote the conditions *alive* and *loaded* to be true (i.e. to hold) at time t, respectively. Then the scenario must satisfy:

$$\begin{array}{l} alive(0) \\ \neg loaded(0) \\ true \to loaded(1) \\ loaded(2) \to \neg alive(3) \end{array}$$

where "¬" stands for the (classical) negation and "→" does for the entailment. As an implicit assumption that $alive(0) \equiv alive(1)$, loading the gun only changes the value of "*loaded*": The conditions do not change unless an action changes them (which is the frame problem).

By the concept of *fluent* (which is a condition to change truth values) (Hanks and McDermott, 1987), we can have one evaluation of the conditions whose changes are minimized:

$$\begin{array}{l} alive(0), alive(1), alive(2), \neg alive(3); \\ \neg loaded(0), loaded(1), loaded(2), loaded(3) \end{array}$$

where another evaluation is not satisfactory, though the changes of the conditions are minimized:

$$\begin{array}{l} alive(0), alive(1), alive(2), \neg alive(3); \\ \neg loaded(0), loaded(1), \neg loaded(2), \neg loaded(3) \end{array}$$

The former sequence of evaluations can be represented in a state-constraint system as follows, where Fn is a function, and $C1$, $C2$ are conditions:

state	0	1	2	3
Fn	*gun-load*	*null*	*shooting*	*null*
$C1$	*alive*	*alive*	*alive*	$\neg$*alive*
$C2$	$\neg$*loaded*	*loaded*	*loaded*	*loaded*

The solution with the state transitions suggests a basic structure of:

$$\begin{array}{lll} \text{state } s & Fn_1 & \text{state } s_1 \\ & Fn_2 & \text{state } s_2 \\ & \dots & \dots \\ & \dots & \dots \\ & Fn_k & \text{state } s_k \end{array}$$

where: (i) Fn_1, Fn_2, ..., Fn_k are terms or functions applicable (with or without conditions), at the state s (conditioned or monitored by some logical formula). (ii) The states s_1, s_2, ..., s_k are regarded as the constraints, respectively, transited from the state s after the application of each of those functions.

3 COMMUNICATION AND FUNCTION APPLICATION

Hennessy-Milner Logic (HML, for short) is conceptually relevant to the state-constraint system. With an action (or a communication) $\langle c \rangle$ as below, formulae as in Hennessy-Milner Logic (HML) are described by the form:

$$\varphi ::= \mathrm{tt} \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle c \rangle \varphi.$$

Following the denotations of formulas, we have extended Hennessy-Milner logic to a multi-modal logic version, based on the papers (Cardelli and Gordon, 2000; Merro and Nardelli, 2005; Hennessy and Milner, 1985; Milner, 1999), for modeling of states monitored in implementation. Possibly with monitorable states as meaning for each formula, we now have the set Φ of formulas, modified from the first version (Yamasaki and Sasakura, 2015):

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

We here have a prefix modality $\langle c \rangle$ (for communication), a postfix one $\rangle t \rangle$ (for function application), a negation (sign) ∼ for incapability of interaction, standard propositions p, the logical negation ¬ and a least fixed operator μ.

The semantics for formulas are definable on the basis of *a transition system*. The transition system (for semantics of logic) 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 $t \in A$ 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.

The reason why 3 assignments of V_{pos}, V_{neg} and V_{inter} are adopted comes from introduction to monitoring *interaction*. 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 $\varphi \in \Phi$.

The Meaning is concerned with two modalities $\langle c\rangle$, $\rangle t\rangle$:

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

(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 φ is positive, and both the intersection "$\cap$" and the subset "$\subseteq$" are componentwise, with assignments of T_{pos} and T_{neg} to x.

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

Modality $\langle c\rangle$ is from communication labelled by c, Modality $\rangle t\rangle$ possibly comes from function applications. When the latter modality is applied to a state s, it may hold a relation $Rel(t)$. It follows that:

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

4 FUNCTIONS REGARDING URL SEARCHING

$\varphi\rangle t\rangle$ with a function t is interpreted as monitoring the state at which the function (a kind of term) is (in interaction mode) applied and possibly transited to the next state.

Upon URL searching, the operations contain unfolding to refine the next references included in the present reference, and folding to return back to the former reference. In this context, we here have some algebraic structure, as a state-constraint system behaviour.

Assume a simple structure of references in URL:

(Home page name) A	
(Contents)	(Reference names) B_1
	B_2
	$\dots$
	$\dots$
	B_k

We then examine two functions of:
(i) unfolding to open the HP (home page) named A to see reference names B_1, …, and B_k, and (ii) folding to close reference names B_1, …, and B_k to have the (HP) name A, as illustrated below.

(Home page name) A	$\to^{unfolding};\ \leftarrow^{folding}$
(Contents)	(Reference names) B_1
	B_2
	$\dots$
	$\dots$
	B_k

The sequence of operations by folding and unfolding can be described in an algebraic structure. To see it, let X be a set of reference (including HP) names, where its power set 2^X contains all the subsets of X, including the emptyset $\emptyset$ and X.

For a function (like unfolding) $f : X \to 2^X$, we have $\hat{f}(x) = \cup_{x \in X}\ f(x)$, such that the function f is extended to the function $\hat{f} : 2^X \to 2^X$. For a function (like folding) $g : 2^X \to X$, we have $\hat{g}(Y) = \{g(Y)\}$, such that the function g is extended to the function $\hat{g} : 2^X \to 2^X$.

We therefore assume the set Ψ of functions of the power set 2^X to the power set 2^X, that is, $2^X \to 2^X$, where each function should supposedly assign the emptyset ($\emptyset \in 2^X$) to the emptyset. The set includes the identity $Id : 2^X \to 2^X$, $Id(Y) = Y$. The function ϕ is assumed to assign the empty set ($\emptyset \in 2^X$) to any $Y \in 2^X$.

The composition $(G \circ F)$ of the functions $F, G \in \Psi$ is defined to be

$$(G \circ F)(Y) = G(F(Y)) \text{ for } Y \in 2^X,$$

where $(G \circ F)$ may be represented by $G \circ F$. The alternation $(F + G) : 2^X \to 2^X$ of $F, G \in \Psi$ is defined to be

$$(F + G)(Y) = F(Y) \cup G(Y),$$

where $(F + G)$ may be represented by $F + G$.

The relation $\equiv$ on the set Ψ is defined:

$$F \equiv G \text{ iff } F(Y) = G(Y) \text{ for any } Y \in 2^X$$

It follows that the relation $\equiv$ is an equivalence relation.

We can see the following properties, which show that $(\Psi, +, \circ, \phi, Id)$ is a semiring:

Proposition 1.

(i) $F + G \equiv G + F$.

(ii) $F + (G + H) \equiv (F + G) + H$.

(iii) $F + \phi \equiv \phi + F \equiv F$.

(iv) $F \circ (G \circ H) \equiv (F \circ G) \circ H$.

(v) $F \circ Id \equiv Id \circ F \equiv F$.

(vi) $F \circ (G + H) \equiv (F \circ G) + (F \circ H)$.

(vii) $(F + G) \circ H \equiv (F \circ H) + (G \circ H)$.

(viii) $F \circ \phi \equiv \phi \circ F \equiv \phi$.

Proof. (i) $(F + G)(Y) = F(Y) \cup G(Y) = G(Y) \cup F(Y) = (G + F)(Y)$ for any $Y \in 2^X$. Thus it holds.
(ii) It can be seen for the same reason as in (i), owing to the associative law in taking the union $\cup$.
(iii) Because $\phi(Y) = \emptyset$ by the definition, it holds.
(iv) $(F \circ (G \circ H))(Y) = F(G(H(Y))) = (F \circ G)(H(Y)) = ((F \circ G) \circ H)(Y)$ for any $F \in 2^X$, as the associative law of function compositions. It therefore hold.
(v) Because *Id* is an identity function on composition, it holds.
(vi) $(F \circ (G + H))(Y) = F(G(Y) \cup H(Y)) = F(G(Y)) \cup F(H(Y)) = ((F \circ G) + (F \circ H))(Y)$. It so holds.
(vii) It is seen by the similar reason of (vi) for this distributive law to hold.
(viii) Because $\phi(Y) = \emptyset$ and $F(\emptyset)$ is defined to be $\emptyset$, this annihilation holds. □

That is, the properties (i), (ii) and (iii) show that $(\Psi, +, \phi)$ is a commutative monoid (a commutative semigroup with the identity ϕ for $+$). The properties (iv) and (v) show that $(\Psi, \circ, Id)$ is a monoid (a semigroup with the identity for $\circ$). (vi) and (vii) are distributive laws, while (viii) is *annihilation* (in a semiring).

For $F \in \Psi$, let $F^* = \cup_{n \in \omega} F^n$, where:

$$F^n = \begin{cases} Id & (n = 0) \\ F^{n-1} \circ F & (n > 0) \end{cases}$$

It follows that $F^* \in \Psi$ such that:

$$F^* = Id + F \circ F^* \equiv Id + F^* \circ F,$$

that is, $(\Psi, +, \circ, \phi, Id)$ is a star semiring.

Semiring with Multiplicative Inverse

The structure may contain the case that

$$\forall F \in \Psi, \exists F' \in \Psi.\ F \circ F' = F' \circ F = Id$$

(where F' is a multiplicative inverse represented by F^{-1}).
(Note) We may take the reduction to have the identity *Id* from $G \circ G^{-1}$ or $G^{-1} \circ G$, until no more reduction could be made, to have an equivalent expression (with respect to "$\equiv$") for a given expression.

Proposition 2. Given an expression with the operations +, $\circ$ and $*$, there is a procedure to reduce it to the equivalent expression (with respect to "$\equiv$") until no more reduction of right or left inverse can be made.

Proof. We can have a mapping $h : EXP \to EXP$ recursively defined as follows, where for an expression $Ex \in EXP$ (the set of expressions with the operations $+$, $\circ$ and $*$) to denote a member in the set Ψ:

$$h(Ex) = \begin{cases} \phi & (Ex = \phi) \\ Id & (Ex = Id) \\ F & (Ex = F) \\ F^{-1} & (Ex = F^{-1}) \\ h(Ex_1) + h(Ex_2) & (Ex = Ex_1 + Ex_2) \\ +_{\langle G_i,\, G_i^{-1} \rangle}\ h(Ex_1/G_i) \circ h(G_i^{-1} \backslash Ex_2) & \\ + +_{\langle H_j^{-1},\, H_j \rangle}\ h(Ex_1/H_j^{-1}) \circ h(H_j \backslash Ex_2) & \\ + h(Ex_1) \circ h(Ex_2) & (E = Ex_1 \circ Ex_2) \\ +_{\langle G_i,\, G_i^{-1} \rangle}\ h(Ex_0^*/G_i) \circ h(G_i^{-1} \backslash Ex_0) & \\ + +_{\langle H_j^{-1},\, H_j \rangle}\ h(Ex_0^*/H_j^{-1}) \circ h(H_j \backslash Ex_0) & \\ + Id + h(Ex_0^*) \circ h(Ex_0) & (Ex = Ex_0^*) \end{cases}$$

with the expressions:

(1) Ex_1/G_i and $G_i^{-1} \backslash Ex_2$ are the right residual and the left residual, respectively.

(2) Ex_1/H_j^{-1} and $H_j\backslash Ex_2$ denote the right residual and the left residual, respectively.

Note that $h(E_0^*/G_i)$, and $h(E_0^*/H_j^{-1})$ are included in $h(E_0^*)$, such that:

$$\begin{aligned}&h(E_0^*/G_i) = \\ &+_{\langle G_k,\ G_k^{-1}\rangle}\ h(Ex_0^*/G_k)\circ h(G_k^{-1}\backslash Ex_0/G_i)\\ &++_{\langle H_l^{-1},\ H_l\rangle}\ h(Ex_0^*/H_l^{-1})\circ h(H_l\backslash Ex_0/G_i)\\ &+h(Ex_0^*)\circ h(Ex_0/G_i),\ \text{and}\end{aligned}$$

$$\begin{aligned}&h(E_0^*/H_j^{-1}) = \\ &+_{\langle G_k,\ G_k^{-1}\rangle}\ h(Ex_0^*/G_k)\circ h(G_k^{-1}\backslash Ex_0/H_j^{-1})\\ &++_{\langle H_l^{-1},\ H_l\rangle}\ h(Ex_0^*/H_l^{-1})\circ h(H_l\backslash Ex_0/H_j^{-1})\\ &+h(Ex_0^*)\cdot h(Ex_0/H_j^{-1}).\end{aligned}$$

A well-known fixed point technique (following S.C. Kleene) is available, for the expression Ex as below to be defined:

$$Ex = Ex_1 + Ex\circ Ex_2 \Rightarrow Ex = Ex_1\circ Ex_2^*$$

such that $h(Ex)$ may be the required expression. □

5 PREDICATES AS APPLIED FUNCTIONS

Bothe positive and negative predicates are thought of as function applications, with reference to formula conditions monitoring states. They may induce relations between states, where the relations are concerned with the operations, union $\cup$ and composition $\circ$. As in interaction mode, we here formulate a postfix modality consisting of the pair as below:

With an assumed predicate set *P-Set*, a par of

(i) a set of sequences of predicates for a collection of sequences of positive conditions, and

(ii) a set of predicates for negative conditions

is to be considered.

Definition 3. Given a set *P-Set*, *P-Set** is the set of all finite sequences of elements from *P-Set*, including the empty sequence ε such that $\varepsilon l = l\varepsilon = l$ for any $l \in$ *P-Set**.

Then a postfix modality $\rangle t\rangle$, where t takes the form $(pseq, neg)$ for $pseq \subseteq$ *P-Set** and $neg \subseteq$ *P-Set*, can be built into the formulas of this paper. Then a relation $Rel(pseq, neg)$ may be defined in the transition system $\mathcal{S}$.

With correspondences and modifications of addition $+$ to union $\cup$ for the relation $Rel(pseq, neg)$ and of multiplication $\bullet$ to concatenation $\circ$ for the relation $Rel(pseq, neg)$, an algebraic structure of some set of pairs $(pseq, neg)$ is below discussed.

For a "consistent" pair of the form $(pseq, neg)$, we make use of:

Definition 4. $l \in pseq$ is *consis* to the set *neg* if any element of l is not in *neg*. The pair $(pseq, neg)$ is *Consis* if any l in *pseq* is *consis* to *neg*.

We now have the algebraic structure: For *Pred*2 $= \{(pseq, neg) \mid pseq \subseteq$ *P-Set** and $neg \subseteq$ *P-Set*, $(pseq, neg)$ is *Consis* $\}$, the structure

$$\langle Pred2, +, \bullet\rangle$$

is to be a *semiring*, where the operations $+$ as addition and $\bullet$ as multiplication are defined, as well as the *star*:

(a) $(pseq_1, neg_1) + (pseq_2, neg_2)$ $=_{def} (pseq_1 \cup pseq_2, neg_1 \cap neg_2)$.

(b)

$$\begin{aligned}&(pseq_1, neg_1)\bullet(pseq_2, neg_2)\\ &=_{def} (pseq_1\cdot pseq_2\\ &\quad -\{l\in pseq_1 | l \text{ is not } consis \text{ to } neg_2\}\cdot pseq_2\\ &\quad - pseq_1\cdot\{l\in pseq_2 | l \text{ is not } consis \text{ to } neg_1\},\\ &\quad neg_1\cup neg_2),\end{aligned}$$

where "$\cdot$" means a concatenation of two (sequence) sets, which provides a new set of sequences obtained by arranging two sequences taken from the two given sets in order.

(c) $(pseq, neg)^\star =_{def} (pseq^*, \emptyset)$ for a star "$\star$", where $pseq^* = \cup_{n\in\omega}\ pseq^n$ (for $pseq^0 = \{\varepsilon\}$ and $pseq^n = pseq^{n-1}\cdot pseq$, $n > 0$, with "$\cdot$" as concatenation of two sets of sequences).

It can be easily observed that:

(i) $\langle Pred2, +, (\emptyset, P\text{-}Set)\rangle$ is a commutative semigroup with the identity $(\emptyset, P\text{-}Set)$, that is , a commutative monoid.

(ii) $\langle Pred2, \bullet, (\{\varepsilon\}, \emptyset)\rangle$ is a semigroup with the identity $(\{\varepsilon\}, \emptyset)$, that is , a monoid.

The we have:

Proposition 5. *The algebraic structure* $\langle Pred2, +, \bullet\rangle$ *is a (star) semiring.*

Proof. (Outline) With observations of two monoids $\langle Pred2, +\rangle$ ($+$: commutative addition) and $\langle Pred2, \bullet\rangle$ ($\bullet$: multiplication), we can see that the multiplication distributes over addition from the left and from the right. Then we have the annihilation that $(\emptyset, P\text{-}Set)\bullet(pseq, neg) = (pseq, neg)\bullet(\emptyset, P\text{-}Set) = (\emptyset, P\text{-}Set)$. It is finally seen for the star operation that:

$$\begin{aligned}&(pseq, neg)^\star\\ &= (\{\varepsilon\}, \emptyset) + (pseq, neg)^\star\bullet(pseq, neg)\\ &= (\{\varepsilon\}, \emptyset) + (pseq, neg)\bullet(pseq, neg)^\star\\ &= (pseq^*, \emptyset).\end{aligned}$$

□

6 CONCLUDING REMARKS

As a conclusion of this positioning, the interactive, functional applications (as in complex AI) may be settled with state transition system concepts, which is also viewed by multi-modal logic with meanings of formulas (conditions). This view has been from a display of AI system solution like Yale Shooting Problem. Then (1) URL searching structures in knowledge-based systems and (2) predicates applicable at states can have been interpreted in algebraic structures. As theories, the view may be relevant to the algebraic informatics as in the classical category theory. The view may be practical in an interactive design for *origami* (Yamasaki and Sasakura, 2015).

As related topics on abstract state machine or state transition system, we have already principles and backgrounds as follows, such that this positioning may be regarded as a refined and original work.
(i) Communication models are well established in relation to action logic with reference to algebraic aspects (Cardelli and Gordon, 2000; Merro and Nardelli, 2005; Milner, 1999) where behavioural sequences may be considered as essential. The modal operator of this positioning may be regarded as relevant to such backgrounds.
(ii) As regards functional programming and actions, semantics are examined (Bertolissi et al., 2006; Mosses, 1992; Reiter, 2001) from operational and declarative methods. This positioning follows an algebraic way different from these semantic views.
(iii) The state-constraint system is relevant to coalgebra (Rutten, 2001; Venema, 2006; Winter et al., 2013; Winter et al., 2015). As regards pushdown store managements, weighted automaton concept (Reps et al., 2005) is known, which is related to a semiring structure with multiplicative (right) inverse of this paper.
(iv) The problem solving methods (Genesereth and Nilsson, 1987; Osorio et al., 2004) are established such that it may have conceived the step by step managements even for action logic (Giordano et al., 2000; van der Hoek et al., 2005). This positioning might present refinements in those directions.

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.

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.

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

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

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

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.

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

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.

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

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. and Sasakura, M. (2015). Multi-modal mu-calculus semantics for knowledge construction. In *Proceedings of 7th IC3K, KEOD*, pages 358–362.