Algebraic Structure of Recursively Constructed References and Its Application to Knowledge Base

Susumu Yamasaki[1][a] and Mariko Sasakura[2][b]

[1] *Okayama University, Tsushima-Naka, Okayama, Japan*

[2] *IPDRE, Tottori University, Hamasaka, Tottori City, Japan*

Keywords: Algebraic Structure, Model Theory, Knowledge Base.

Abstract: From management views on complex website page structures, we formulate an algebraic structure of recursively constructed page references as presenting situations of them to the website with 3-valued domain. Algebraic structure of references, abstracted from website page references, is here expressed as a finite or countably infinite set of rules, where each rule is defined, by representing the recursive relations among web page references. The situations of a reference with request to the website can be denoted as the acquisitive positive, rejective negative and suspended negative, respectively. With respect to algebraic structure, a fixed point of the mapping associated with the rule set may be a model denoting consistent evaluations to assign the situations of references to 3-valued domain. Model theory for representation of consistent evaluations of references and the rule set (constructed with references) is newly settled if a fixed point consistently exists. A retrieval derivation to detect acquisitive positives and rejective negatives can be presented, to be sound with respect to the model, based on the inference by negation as failure, which is related to the suspended negative. As multiple knowledge base formed by a tuple of rule sets, this paper next presents algebraic structure of a distributed knowledge system constrained by a state, and sequential applications of such systems, containing state transitions. Model theory can be defined with fixed point of the mapping associated with the distributed knowledge system, although the fixed point may not be always applied to modeling. If consistent fixed point modeling is available, we may have a model of the distributed knowledge system, constrained by a state. Then the application of such a distributed knowledge system may be considered as causing state transitions, following modeling and designed state transitions.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The well organized website contains web pages so that complex information system may be constructed. In this paper, we study the algebraic structure of web page references, in more details, from the purpose of analyzing the page references recursively constructed among references, whose situations are in the cases of being active, inactive or different, in close relation of a reference request to the website. The active, inactive and different situations are then interpreted as taking acquisitive, rejective and suspended values (which are, as algebraic elements, assigned to references), such that algebraic element may be knowledgeable by itself. In this context, the algebraic structure of recursively constructed references are formulated with 3-valued domain. The structure is complex enough, however, references recursion may be compact to analyze and to apply to knowledge base with processing. The 3-domain is defined by the set of truth, falsity and the unknown, for the acquisitive, rejective and suspended values, respectively. What values are to be consistent assignments of references in a given algebraic structure is discussed, by the method of evaluation of references and the whole structure, as a model theory. We newly present a fixed point model for the algebraic structure, to make it clearer for the complex structure of references to be captured as consistent in the recursive construction of references. Based on the evaluation of each reference in a 3-valued domain, the model theory is regarded as relevant to consistency of data integrity, when the complex structure of references is applied to knowledge base.

For model theory, retrieval derivation to the structure with rejective negation is also presented, as knowledge processing. With reference to knowledge

[a] https://orcid.org/0000-0001-7895-5040

[b] https://orcid.org/0000-0002-8909-9072

Yamasaki, S. and Sasakura, M.
Algebraic Structure of Recursively Constructed References and Its Application to Knowledge Base.
DOI: 10.5220/0012544500003708
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 9th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2024)*, pages 83-90
ISBN: 978-989-758-698-9; ISSN: 2184-5034
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

processing, we may see the algebraic structure, from the view of knowledge base with logic.

About logic for knowledge and reasoning, we have seen the backgrounds:
(a) The propositional system and algebra for inquisitive logic with negative variants are organized, by means of intermediate logic in the class between those classes of inuitionistic and propositional calculi. (Bezhanishvili et al., 2022). Nonmonotonic logic on reasoning and inference mechanism should contain applicative aspects with respect to design and programming (Hanks and McDermott, 1987; Reiter, 2001).
(b) Modal logic contains semantics based on state space in Kleene-Kripke theory. Not only propositional logic but also first-order modal logic is formally dealt with (Fitting, 2002). With reference to knowledge processing and computability, quantified modal logic is discussed (Rin and Walsh, 2016). For stepwise removal of objects, there is dynamic modality (Bentham et al., 2022). Two-dimensional modality is studied (Gessel, 2022).
(c) With respect to representation of implementation, dynamic logic for action was presented (Spalazzi and Traverso, 2000). As other dynamic aspects of knowledge processing, relevant works are made with the concepts of justified belief truth (Egre et al., 2021), reduction of decidability (Rasga et al., 2021) and inference theory (Tennant, 2021).

Compared with the backgrounds, this paper is concerned with knowledge base, which the algebraic structure of references denotes, as well as knowledge retrieval sound with respect to fixed point model. The primary aim of this paper is relevant to nonmonotonic logic. We next obtain the algebraic structure of a distributed knowledge system containing a tuple of knowledge bases, where the whole system is constrained by a state (environment). After knowledge processing to each knowledge base, state transitions may be supposed for further knowledge processing in next states, where each state constrains a distributed knowledge system.

Revising the work (S.Yamasaki et al., 2023), we take knowledge processing with common negatives in a state, constraining a distributed knowledge system. On condition that only negatives may be common, consistency in the whole distributed knowledge system is treated, without explicit communications among knowledge bases. The state transitions are characterized as sequential applications of state constrained systems.

We here note the literatures concerned with behaviours of distributed systems, paying attention to the traverses of states in a system: Sequences traversing states in a distributed system may be closely related to the method of automata (Droste et al., 2009). As regards varying or moving processes, formulations have been developed in mobility of ambients (Cardelli and Gordon, 2000; Merro and Nardelli, 2005).

This paper captures the environment just by a state name. About the traverse of states caused by applications of knowledge processing, we have semantics for the effect of the sequential applications, on the basis of algebraic semiring.

The paper is organized as follows. Section 2 formally provides a rule set as algebraic structure of recursively constructed references. In Section 3, the meaning (model) of the algebraic structure may be newly defined by fixed point of a mapping associated with the rule set. Section 4 presents the retrieval derivation from the algebraic structure with respect to its model. In Section 5, model theory of a distributed knowledge system with common negatives is developed, as application of the rule sets to knowledge bases and to a distributed knowledge system. Section 5 is also concerned with the traverses of states, with the model theory of the distributed systems constrained by states. Concluding remarks are given in Section 6.

2 RECURSIVELY CONSTRUCTED REFERENCES

As knowledge acquisition means, the website is established. We note the static structure constructed by organizing the website references of web pages. It is concerned with recursive construction, as well as with negatives owing to the situations of the website.

Algebraic Structure of References
As is seen, we observe web page reference with request to some website, as classified into several situation sets. In this paper, we suppose 3 situation sets of the website collecting pages with references to them:

(a) The website is active so that the reference to the site may be considered as in an acquisitive set (for knowledge).

(b) The website is inactive so that the reference may be regarded as in a rejective set.

(c) The website (to which the reference is made) is different from the one which the reference should be settled to, such that the reference may be interpreted as in a suspended set.

Figure 1 presents views on the situation sets.

We also observe the relation among references where the referenced page may contain finitely many

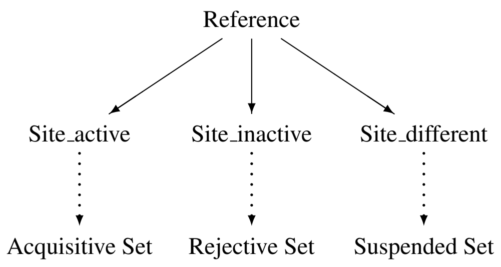

Figure 1: Web page reference is classified.

page references, as likely denoted by:

$$\langle Ref_1, \ldots, Ref_n \rangle \text{ (contained in) } Ref,$$

where "*Ref*" is a reference considered recursively constructed with references Ref_1, ..., Ref_n ($n \geq 0$). From application views, the reference is knowledgeable, as well, in the sense that the reference may recursively contain references.

We take *Ac*, *Re* and *Sus* as the acquisitive set, the rejective set and the suspended set of references, respectively, with *Knowl* as knowledge containing a reference. Making use of a rule of the form

$$(Ac, Re, Sus)) \rhd Knowl,$$

and of a (finite or countably infinite) set of rules, we study the meaning of algebraic structure of rule set. Its model theory is concerned with knowledge acquirements.

Algebraic Rule
We firstly assume a domain A as a set of references (as knowledgeable objects). We then make use of:

$$A^{not} = \{not\ a \mid a \in A\},$$
$$A^{\sim} = \{\sim a \mid a \in A\}.$$

where (i) *not a* is to be interpreted as rejection of a (or its negation as in Heyting algebra), and (ii) $\sim a$ is as suspension of a, exactly defined later in model theory. .A rule is of the form r, in a formal way as represented by acquisitive, rejective and suspended informations contained in an acquired one. It is regarded as knowledge (with the notations "$\emptyset$" and "$|$" as the empty set and "or", respectively), from the application views as in Section 5, denoted by the form as follows.

$r = (Ac, Re, Sus) \rhd Knowl$, where for $a \in A$:

$$\begin{aligned} & Ac \subseteq A, Re \subseteq A^{not}, Sus \subseteq A^{\sim} \text{ such that} \\ & Ac ::= \emptyset \mid \{a\} \cup Ac \\ & Re ::= \emptyset \mid \{not\ a\} \cup Re \\ & Sus ::= \emptyset \mid \{\sim a\} \cup Sus \\ & Knowl ::= a \mid not\ a \mid \sim a \end{aligned}$$

That is, a rule is of the form $(Ac, Re, Sus) \rhd Knowl$ where: (a) $Ac \subseteq A$ is a finite acquisitive set. (b) $Re \subseteq A^{not}$ is a finite rejective set. (c) $Sus \subseteq A^{\sim}$ is a finite suspended set. (d) $Knowl = a$, *not a* or $\sim a$ is acuired knowledge (for $a \in A$).

The expression $(Ac, Re, Sus) \rhd a$, $(Ac, Re, Sus) \rhd not\ a$, or $(Ac, Re, Sus) \rhd \sim a$ may be adopted, if the form *Knowl* is $a \in A$ or $not\ a \in A^{not}$ or $\sim a \in A^{\sim}$, respectively.

R is inductively defined as a (finite or countably infinite) set of rules. It is regarded as knowledge base, as in Section 5: $R ::= \emptyset \mid r \cup R$.

As a denotation of the page reference, the rule form $(\cup_i (Ac_i, Re_i, Sus_i)) \rhd Knowl$ may be considerable, however, it can be replaced by a rule set $\cup_i \{(Ac_i, Re_i, Sus_i) \rhd Knowl\}$.

3 MODEL THEORY IN ALGEBRAIC STRUCTURE

As a more general case than the one (Yamasaki and Sasakura, 2023) and the conditional causal relation (Yamasaki and Sasakura, 2021), we examine the meaning of the rule set defined in Section 2. While many-valued logic provability is presented (Pawlowski and Urbaniak, 2018), answer set programming and problem solving are compiled in 2-valued logic for practice (Gebser and Schaub, 2016; Kaufmann et al., 2016).

The rejection negation is regarded as negation in Heyting algebra, as well as in intuitionistic propositional logic. For the suspended negation, one more negation is to be defined, so that 3-valued evaluation is of use, with the unknown (as a value). But we do not deal with weak negation as in the case (Yamasaki, 2006). With a 3-valued domain, we present fixed point modeling, based on a mapping associated with the rule set.

3.1 3-valued Domain

A bounded lattice $K = (\{f, unk, t\}, \bigvee, \bigwedge, \bot, \top)$ equipped with the partial order $\sqsubseteq$ and an implication $\Rightarrow$ may be taken:
(a) $\bot$ and $\top$ are the least and the greatest elements f and t, respectively, with respect to the partial order $\sqsubseteq$ such that $\bot = f \sqsubseteq unk \sqsubseteq t = \top$.
(b) The least upper bound (*join* with $\bigvee$) and the greatest lower bound (*meet* with $\bigwedge$) exist for any two elements of the set $\{f, unk, t\}$.
(c) The implication $\Rightarrow$ (a relation on the set $\{f, unk, t\}$) is defined in a way that $z \sqsubseteq (x \Rightarrow y)$ iff $x \bigwedge z \sqsubseteq y$.

t is the truth, f is the falsity and *unk* (the unknown as t or f) is the undefined for the truth value, where

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

the partial order is defined, regarding the truth value.

Evaluation for Rule Set
A valuation $V : A \to \{f, unk, t\}$ is assumed with the bounded lattice K. With respect to V, the value $eval_V(E)$ of an expression E is given.

$eval_V(a) = V(a)$ $(a \in A)$
$eval_V(not\ a)$
$=$ if $(eval_V(a) = f)$ then t else f
$eval_V(\sim a)$
$=$ if $(eval_V(a) = f)$ then t
else if $(eval_V(a) = unk)$ then unk else f
$eval_V(Knowl)$
$=$ if $(Knowl = a)$ then $eval_V(a)$
else if $(Knowl = not\ a)$ then $eval_V(not\ a)$
else if $(Knowl = \sim a)$ then $eval_V(\sim a)$

$eval_V(Ac) = \bigwedge_{a \in Ac} V(a)$
$eval_V(Re) = \bigwedge_{not\ a \in Re} eval_V(not\ a)$
$eval_V(Sus) = \bigwedge_{\sim a \in Sus} eval_V(\sim a)$
$eval_V((Ac, Re, Sus))$
$= eval_V(Ac) \bigwedge eval_V(Re) \bigwedge eval_V(Sus)$

$eval_V((Ac, Re, Sus) \rhd Knowl)$
$=$ if $eval_V((Ac, Re, Sus)) \sqsubseteq eval_V(Knowl)$
then t else if $(eval_V(Knowl) = f)$ then f else unk

$eval_V(R) =$
$\bigwedge_{((Ac,Re,Sus) \rhd Knowl) \in R} eval_V((Ac, Re, Sus) \rhd Knowl)$
($\bigwedge$ operates on a countable set)

Note that $eval_V(\emptyset) = t$ for Ac, Re, Sus and R to be the empty set $\emptyset$.

Model of Rule set
For a pair $(I, J) \in 2^A \times 2^A$ such that $I \cap J = \emptyset$, i.e., I and J are disjoint, the valuation

$$V(I, J) : A \to \{f, unk, t\}$$

is defined to be:

$V(I,J)(a)$
$=$ if $a \in I$ then t else if $a \in J$ then f
else unk

such that $eval_{V(I,J)}(R)$ may be settled inductively as above. If the disjoint pair (I, J) $(I \cap J = \emptyset)$ causes $eval_{V(I,J)}(R) = t$, the pair is called a model of the rule set R.

3.2 Model of Rule Set with Fixed Point Theory

For a method to obtain a model of the given rule set, a fixed point of the mapping associated with the rule set may be taken, although we cannot always have such a fixed point, because of nonmonotonic functionality of the mapping.

Given a rule set R (based on the set A of objects), for each rule $(Ac, Re, Sus) \rhd a$ $(a \in A)$ to be evaluated as t (following the definition of the evaluation $eval_V$ with the valuation V) such that $eval_V(R) = t$, we examine mutually exclusive cases. For each $(Ac_1, Re_1, Sus_1) \rhd not\ \ b$, or $(Ac_2, Re_2, Sus_2) \rhd \sim c$ in R, we can exhaustively examine the cases for $(Ac_1, Re_1, Sus_1) \rhd b$ or $(Ac_2, Re_2, Sus_2) \rhd c$.

From such an inspection, a pair (I, J) for the valuation $V(I, J)$ to model the rule set R is to be given as a fixed point of the formally defined mapping $Trans_R$ (as below). The pair $(I, J) = Trans_R(I, J)$ (by Definition 1) may be taken (to the valuation $V(I, J)$), as satisfactory.

Definition 1. Given a rule R, a mapping $Trans_R : 2^A \times 2^A \to 2^A \times 2^A$ is defined to be

$$Trans_R(I, J) = (I', J')$$

such that the pair (I', J') may be given:

(a) if $(\forall (Ac, Re, Sus) \rhd a \in R.\ ((\exists b \in Ac.\ b \in J)$ or $(\exists not\ c \in Re.\ c \notin J)$ or $(\exists \sim d \in Sus.\ d \in I)))$, then $a \in J'$
(b) else if $(\exists (Ac, Re, Sus) \rhd a \in R.$ $((\forall b \in Ac.\ b \in I)$ and $(\forall not\ c \in Re.\ c \in J)$ and $(\forall \sim d \in Sus.\ d \in J)))$ on condition that $(\forall (Ac_1, Re_1, Sus_1) \rhd not\ a \in R.$ $((\exists b \in Ac_1.\ b \in J)$ or $(\exists not\ c \in Re_1.\ c \notin J)$ or $(\exists \sim d \in Sus_1.\ d \in I)))$ and $(\forall (Ac_2, Re_2, Sus_2) \rhd \sim\ a \in R.$ $((\exists b \in Ac_2.\ b \in J)$ or $(\exists not\ c \in Re_2.\ c \notin J)$ or $(\exists \sim d \in Sus_2.\ d \in I)))$, then $a \in I'$
(c) else if a rule of the form $((Ac, Re, Sus) \rhd a)$ exists in R and $(\forall (Ac, Re, Sus) \rhd a \in R.$ $((\exists b \in Ac.\ b \notin I)$ or $(\exists not\ c \in Re.\ c \notin J)$ or $(\exists \sim d \in Sus.\ d \notin J)))$ and $(\forall (Ac_1, Re_1, Sus_1) \rhd not\ a \in R.$ $((\exists b \in Ac_1.\ b \in J)$ or $(\exists not\ c \in Re_1.\ c \notin J)$ or $(\exists \sim d \in Sus_1.\ d \in I)))$ and $(\forall (Ac_2, Re_2, Sus_2) \rhd \sim\ a \in R.$ $((\exists b \in Ac_2.\ b \notin I)$ or $(\exists not\ c \in Re_2.\ c \notin J)$ or $(\exists \sim d \in Sus_2.\ d \notin J))))$, then $a \notin I' \cup J'$
else undefined

If $Trans_R(I, J) = (I, J)$ such that $I \cap J = \emptyset$, the pair is called a consistent fixed point of $Trans_R$.

Proposition 2. *If (I, J) is a consistent fixed point of $Trans_R$ then $eval_{V(I,J)}(R) = t$.*

Proof. For each $a \in A$, we examine the evaluation of all the related rules in R with respect to the valuation $V(I, J)$.

(i) When $a \in I$,

$$(\forall (Ac, Re, Sus) \triangleright a \in R. \; eval_{V(I,J)}((Ac, Re, Suc) \triangleright a) = t).$$

Because $a \in I$,

$(\forall (Ac_1, Re_1, Sus_1) \triangleright not\ a \in R.$
$((\exists b \in Ac_1.\ b \in J)$ or $(\exists not\ c \in Re_1.\ c \notin J)$ or
$(\exists \sim d \in Sus_1.\ d \in I))).$

It follows that

$eval_{V(I,J)}(Ac_1) = f$ or $eval_{V(I,J)}(Re_1) = f$ or
$eval_{V(I,J)}(Sus_1) = f$.

Thus $eval_{V(I,J)}((Ac_1, Re_1, Sus_1) \triangleright not\ a) = t$.
As is the similar case, for $a \in I$,

$(\forall (Ac_2, Re_2, Sus_2) \triangleright \sim a \in R.$
$((\exists b \in Ac_2.\ b \in J)$ or $(\exists not\ c \in Re_2.\ c \notin J)$ or
$(\exists \sim d \in Sus_2.\ d \in I))).$

Then

$eval_{V(I,J)}(Ac_2) = f$ or $eval_{V(I,J)}(Re_2) = f$ or
$eval_{V(I,J)}(Sus_2) = f$.

Finally $eval_{V(I,J)}((Ac_2, Re_2, Sus_2) \triangleright \sim a) = t$.
(ii) Assume that $a \in J$. Then

$(\forall (Ac, Re, Sus) \triangleright a \in R.$
$((\exists b \in Ac.\ b \in J)$ or $(\exists not\ c \in Re.\ c \notin J)$ or
$(\exists \sim d \in Sus.\ d \in I))).$

Therefore

$eval_{V(I,J)}(Ac) = f$ or $eval_{V(I,J)}(Re) = f$ or
$eval_{V(I,J)}(Sus) = f$.

It causes $eval_{V(I,J)}((Ac, Re, Sus) \triangleright a) = t$. On the other hand, $eval_{V(I,J)}(not\ a) = t$ such that

$$eval_{V(I,J)}((Ac_1, Re_1, Sus_1) \triangleright not\ a) = t$$

for any $(Ac_1, Re_1, Sus_1) \triangleright not\ a$ in R.
Also we have $eval_{V(I,J)}(\sim a) = t$ such that

$$eval_{V(I,J)}((Ac_2, Re_2, Sus_2) \triangleright \sim a) = t$$

for any $(Ac_2, Re_2, Sus_2) \triangleright \sim a$ in R.
(iii) In case that $a \notin I \cup J$: there is a rule of the form $(Ac, Re, Sus) \triangleright a \in R$ such that $eval_{V(I,J)}(a) = unk$, where $eval_{V(I,J)}((Ac', Re', Sus')) = unk$ or f for any rule of the form $(Ac', Re', Sus') \triangleright a$. It follows that $eval_{V(I,J)}((Ac', Re', Sus') \triangleright a) = t$ for any rule of the form $(Ac', Re', Sus') \triangleright a$, on condition that

$(\forall (Ac_1, Re_1, Sus_1) \triangleright not\ a \in R.$
$((\exists b \in Ac_1.\ b \in J)$ or $(\exists not\ c \in Re_1.\ c \notin J)$ or
$(\exists \sim d \in Sus_1.\ d \in I)))$ and
$(\forall (Ac_2, Re_2, Sus_2) \triangleright \sim a \in R.$
$((\exists b \in Ac_2.\ b \notin I)$ or $(\exists not\ c \in Re_2.\ c \notin J)$ or
$(\exists \sim d \in Sus_2.\ d \notin J))).$

From the above condition: for any rule of the form

$$(Ac_1, Re_1, Sus_1) \triangleright not\ a \in R,$$

$eval_{V(I,J)}((Ac_1, Re_1, Sus_1)) = f$, while
$eval_{V(I,J)}(not\ a) = f$. Thus

$$eval_{V(I,J)}((Ac_1, Re_1, Sus_1) \triangleright not\ a) = t.$$

For any rule of the form

$$(Ac_2, Re_2, Sus_2) \triangleright \sim a \in R,$$

$eval_{V(I,J)}((Ac_2, Re_2, Sus_2)) = unk$ or f, while
$eval_{V(I,J)}(\sim a) = unk$. Thus

$$eval_{V(I,J)}((Ac_2, Re_2, Sus_2) \triangleright \sim a) = t.$$

□

4 RETRIEVAL DERIVATION IN RULE SET

We conceive model theory in some case of replacement of the rejective negation *not* by the suspended negation $\sim$. Negation as failure derivation (which corresponds to the suspended negation $\sim$) may be extended to be applicable to the model theory (in Section 3). In the Japanese poem analysis (Yamasaki and Yokono, 2008), this idea was adopted, while the analysis may be refined by the following derivation.

Given a rule set R, we have the recursively defined derivation. Its implementation is applicable to retrievals, when the algebraic rule set may be regarded as knowledge base. Just a procedural aspect is now presented, with a variable G ranging over the set $A \cup A^{not} \cup A^{\sim}$, such that $G : suc$ and $G : fail$ may stand for succeeding and failing derivations (of retrieval), respectively.

(1) $\emptyset : suc$.

(2) When there is $(Ac, Re, Sus) \triangleright a \in R$ such that $(\forall (Ac_1, Re_1, Sus_1) \triangleright not\ a \in R.\ Ac_1 \cup Re_1 \cup Sus_1 : fail)$ and $(\forall (Ac_2, Re_2, Sus_2) \triangleright \sim a \in R.\ Ac_2 \cup Re_2 \cup Sus_2 : fail)$, with $(Ac \cup Re \cup Sus \cup G : suc)$, then $\{a\} \cup G : suc$.

(3) If $\{a\} : fail$ and $G : suc$ then $\{not\ a\} \cup G : suc$.

(4) If $\{a\} : fail$ and $G : suc$ then $\{\sim a\} \cup G : suc$.

(5) If $(\forall (Ac, Re, Sus) \triangleright a \in R.\quad (Ac \cup Re \cup Sus : fail))$, then $\{a\} : fail$.

(6) If $\{a\} : fail$ then $\{a\} \cup G : fail$.

(7) If $\{a\} : suc$ then $\{not\ a\} \cup G : fail$.

(8) If $\{a\} : suc$ then $\{\sim a\} \cup G : fail$.

The above retrieval derivation may contain soundness with respect to some model (I, J) of R, in the sense:

Proposition 3. *Assume that for a rule set R, (I,J) is a consistent fixed point of $Trans_R$. For the derivation to R, we have:*

(1) *If $\{a\} : suc$, then $a \in I$.*
(2) *If $\{a\} : fail$, then $a \in J$.*

Proof. (1) If $\{a\} : suc$, then

$$\forall (Ac_1, Re_1, Sus_1) \rhd not\ a \in R.\ Ac_1 \cup Re_1 \cup Sus_1 : fail,$$

where
$(\exists b_1 \in Ac_1.\{b_1\} : fail)$ or
$(\exists not\ c_1 \in Re_1.\{not\ c_1\} : fail)$
(i.e., $\exists not\ c_1 \in Re_1.\{c_1\} : suc$) or
$(\exists {\sim} d_1 \in Sus_1.\ \{{\sim} d_1\} : fail)$
(i.e., $\exists {\sim} d_1 \in Sus_1.\ \{d_1\} : suc$).
By induction hypothesis, there is $b_1 \in J$ or $c_1 \in I$ or $d_1 \in I$, for the assumed derivation $\{a\} : suc$. Also

$$\forall (Ac_2, Re_2, Sus_2) \rhd {\sim} a \in R.\ Ac_2 \cup Re_2 \cup Sus_2 : fail,$$

where
$(\exists b_2 \in Ac_2.\{b_2\} : fail)$ or
$(\exists not\ c_2 \in Re_2.\{not\ c_2\} : fail)$
(i.e., $\exists not\ c_2 \in Re_2.\{c_2\} : suc$) or
$(\exists {\sim} d_2 \in Sus_2.\ \{{\sim} d_2\} : fail)$
(i.e., $\exists {\sim} d_2 \in Sus_2.\ \{d_2\} : suc$).
By induction hypothesis, there is $b_2 \in J$ or $c_2 \in I$ or $d_2 \in I$, for the assumed derivation $\{a\} : suc$.

2 cases are also to be examined:
(a) If $(\emptyset, \emptyset, \emptyset) \rhd a \in R$, as a basis, it is evident that $a \in I$.
(b) If $(Ac, Re, Sus) \rhd a \in R$ such that

$$Ac \cup Re \cup Sus : suc,$$

as induction step, we can reason that
$(\forall b \in Ac.\ b \in I)$ and $(\forall not\ c \in Re.\ c \in J)$
and $(\forall {\sim} d \in R.\ d \in J)$,
from the condition that $(\forall b \in Ac.\ \{b\} : suc)$
and $(\forall not\ c \in Re.\ \{not\ c\} : suc)$
and $(\forall {\sim} d \in Sus.\{{\sim} d\} : suc)$.
This completes the induction step, such that $a \in I$.
(2) If $\{a\} : fail$, then

$$(\forall (Ac, Re, Sus) \rhd a \in R.\ (Ac \cup Re \cup Sus : fail)).$$

When there is no rule of the form $(Ac, Re, Sus) \rhd a$, then $a \in J$. Otherwise, it follows that
$(\exists b \in Ac.\{b\} : fail)$ or
$(\exists not\ c \in Re.\ \{not\ c\} : fail)$
(i.e., $\exists not\ c \in Re.\ \{c\} : suc$) or
$(\exists {\sim} d \in Sus.\ \{{\sim} d\} : fail)$
(i.e., $\exists {\sim} d \in Sus.\ \{d\} : suc$),
which are respectively caused by $(\exists b \in Ac.\{b\} : fail)$ or $(\exists not\ c \in Re.\ \{c\} : suc)$ or $(\exists {\sim} d \in Sus.\ \{d\} : suc)$. By induction, $b \in J$ or $c \in I$ or $d \in I$. This concludes that $a \in J$. □

5 MULTIPLE KNOWLEDGE BASE CONSTRAINED BY STATE

A distributed knowledge system, constrained by a state, is formulated as multiple knowledge base (a tuple of the algebraic rule sets as above described). We suppose that the negatives (rejective negations) from each rule set is common. We then have the meaning of multiple knowledge base with common negatives, by presenting a treatment of fixed point theory. The state transitions may be caused by sequential applications of state constrained, distributed knowledge systems, where nondeterministic choices of the next state transition are admissible.

Distributed Knowledge System
With an assumed state set S, and a variable s over the set S, we have got a formal definition of the state constraint, distributed knowledge system (of multiple knowledge base) DK containing the rule sets.

$$DK ::= (s)\Sigma$$
$$\Sigma ::= null \mid R[s]; \Sigma$$

where R is defined in Section 2 and ";" (semicolon) means the concatenation, with the empty list $null$.

The list (sequence) Σ may be alternatively denoted by the form of tuple $\Sigma = \langle R_1[s_1], \ldots, R_n[s_n] \rangle$ $(n \geq 0)$, where $(s)\Sigma$ may be dealt with, as follows.
(a) Σ is a distributed knowledge system consisting of rule sets $R_1, \ldots, R_n$. If $n = 0$ then $\Sigma = null$ (as the empty tuple). s is a constraint state of $(s)\Sigma$.
(b) A fixed point method with the models of $R_1, \ldots, R_n$ may be definable, on condition that the negatives (rejective negations) should be common among the models.
(c) s_i as in $R_i[s_i]$ indicates the next state, with some model as an operation on the rule set R_i and after the operation.

For a distributed knowledge system (with a constraint state s), $\Sigma = \langle R_1[s_1], \ldots, R_n[s_n] \rangle$, we may have models of $R_1, \ldots, R_n$, such that negatives are common. A tuple of rule sets $\sigma = \langle R_1, \ldots, R_n \rangle$ $(n \geq 1)$ is adopted, in correspondence to Σ. To get models of a distributed knowledge system, we extend the method (in Section 3) of how each rule set is denoted.

A valuation $V_i : A \to \{f, unk, t\}$ $(1 \leq i \leq n)$ is assumed with the bounded lattice K, as in the case of Section 3. The evaluations of R_k are to be executed for $1 \leq k \leq n$.

Evaluation and Model for Rule Sets
With $(V_1, \ldots, V_n)$, the value $Eval_{(V_1, \ldots, V_n)}(\sigma)$ is given

by:

$$\begin{aligned}&Eval_{(V_1,\ldots,V_n)}(\sigma)\\ &= Eval_{(V_1,\ldots,V_n)}(\langle R_1,\ldots,R_n\rangle)\\ &= (eval_{V_1}(R_1),\ldots,eval_{V_n}(R_n))\end{aligned}$$

For a pair $(I,J) \in 2^A \times 2^A$ such that $I \cap J = \emptyset$, i.e., I and J are disjoint, the valuation

$$V_k(I,J) : A \to \{f, unk, t\}\ (1 \le k \le n)$$

is used as $V(I,J)$ in Section 3.

If with the disjoint pairs (I_k,J_k) $(1 \le k \le n)$ for a tuple σ of rule sets,

$$Eval_{(V_1(I_1,J_1),\ldots,V_n(I_n,J_n))}(\sigma) = (t,\ldots,t)$$

then the pairs are called a model of σ. Just taking a direct product of pairs each of which may be defined by the mapping of $Trans_{R_k}$ $(1 \le k \le n)$, we have a mapping of Tr_σ.

Mapping Associated with Tuple of Rule Sets
Given a tuple of rule sets $\sigma = \langle R_1,\ldots,R_n\rangle$, a mapping

$$Tr_\sigma : (2^A \times 2^A)^n \to (2^A \times 2^A)^n$$

is defined to be

$$\begin{aligned}&Tr_\sigma((I_1,J_1),\ldots,(I_n,J_n))\\ &= (Trans_{R_1}(I_1,J_1),\ldots,Trans_{R_n}(I_n,J_n))\\ &= ((I'_1,J'_1),\ldots,(I'_n,J'_n))\end{aligned}$$

such that each pair (I'_k,J'_k) is given for the pair (I_k,J_k) with $Trans_{R_k}$, where $Trans_{R_k}$ follows Definition 1.

Following Proposition 2, the fixed point of the mapping Tr_σ may be possibly obtained such that each R_k $(1 \le k \le n)$ is modeled, with the condition that negatives are common:

Proposition 4. *If*

$$Tr_\sigma((I_1,J),\ldots,(I_n,J)) = ((I_1,J),\ldots,(I_n,J))$$

such that $I_k \cap J = \emptyset$ $(1 \le k \le n)$ *then*

$$\begin{aligned}&Eval_{(V(I_1,J),\ldots,V(I_n,J))}(\sigma) = (t,\ldots,t),\\ &i.e., eval_{V(I_k,J)}(R_k) = t\ (1 \le k \le n).\end{aligned}$$

Proof. For each R_k, the proof of Proposition 2 can be applied, where each $Trans_{R_k}$ is independent of another $Trans_{R_j}$ $(k \ne j)$, except that the set J is common. Tr_σ is defined in terms of such mutually independent $Trans_{R_k}$. Therefore the above fixed point of Tr_σ is a tuple of the consistent fixed points of $Trans_{R_k}$ $(1 \le k \le n)$. Thus the above fixed point of Tr_σ is associated with $Eval_{(V(I_1,J),\ldots,V(I_n,J))}(\sigma)$, which is a tuple of n values of t. □

The fixed points (I_k,J) $(1 \le k \le n)$ are components of the (tuple) fixed point of σ, in:

$$\Sigma = \langle R_1[s_1],\ldots,R_n[s_n]\rangle\ (n \ge 1)$$

With (I_k,J) $(1 \le k \le n)$ as models in Σ, we may design question answering on the model (I_k,J) of the rule set R_k of Σ, and the state transition to s_k for $1 \le k \le n$. We have nondeterministic choices to s_k by (I_k,J), $(I_k,J) : s \mapsto s_k$ for $1 \le k \le n$. With respect to models (I_k,J) $(1 \le k \le n)$, the distributed derivation (Yamasaki and Sasakura, 2023) and its soundness can be corrected and revised to the derivations working for rule sets R_k $(1 \le k \le n)$:

(1) $\emptyset : k\text{-}suc$.

(2) When there is $(Ac,Re,Sus) \rhd a \in R_k$ such that $(\forall(Ac_1,Re_1,Sus_1) \rhd not\ \ a \in R_k.\ Ac_1 \cup Re_1 \cup Sus_1 : fail)$ and $(\forall(Ac_2,Re_2,Sus_2) \rhd {\sim}a \in R_k.\ Ac_2 \cup Re_2 \cup Sus_2 : fail)$, with $(Ac \cup Re \cup Sus \cup G : k\text{-}suc)$, then $\{a\} \cup G : k\text{-}suc$.

(3) If $\{a\} : fail$ and $G : k\text{-}suc$ then $\{not\ \ a\} \cup G : k\text{-}suc$.

(4) If $\{a\} : fail$ and $G : k\text{-}suc$ then $\{{\sim}a\} \cup G : k\text{-}suc$.

(5) If, for any $1 \le j \le n$, $(\forall(Ac,Re,Sus) \rhd a \in R_j.\ (Ac \cup Re \cup Sus : fail))$, then $\{a\} : fail$.

(6) If $\{a\} : fail$ then $\{a\} \cup G : fail$.

(7) If $\{a\} : k\text{-}suc$ then $\{not\ \ a\} \cup G : fail$.

(8) If $\{a\} : k\text{-}suc$ then $\{{\sim}a\} \cup G : fail$.

Meaning of Distributed Knowledge Systems
The distributed knowledge system (with a constraint state) DK is to denote the set of objects (of the form) $(s)\Sigma$. DK^* stands for the set of finite sequences from DK including the empty sequence λ. The sequence of DK^* may denote a finite number of sequential applications of the objects (possibly considered as processes) in DK, or a finite number of transition sequences of objects in DK. The semantics for the sequence in DK^* is defined inductively as follows, for a state set S and its power set 2^S:

$$Sem : DK^* \to (S \to 2^S),$$

such that for a state $s \in S$,

$$Sem[\![(s')null]\!](s) = \emptyset \text{ (with any } s' \in S)$$

$$Sem[\![\lambda]\!](s) = \{s\}$$

$$Sem[\![(s')\Sigma]\!](s) = \begin{cases} \{s_1,\ldots,s_n\} & \text{if } (s = s') \text{ and } (\exists(I_k,J). \\ & (I_k,J) \text{ is a model of } R_k) \\ & \text{for any } k\ (1 \le k \le n) \text{ in} \\ & \Sigma = \langle R_1[s_1],\ldots,R_n[s_n]\rangle \\ \emptyset & \text{otherwise}\end{cases}$$

$$\begin{aligned}&Sem[\![Xy]\!](s)\\ &= \cup_{t \in Sem[\![X]\!](s)}\ Sem[\![y]\!](t)\ (X \in DK, y \in DK^*)\end{aligned}$$

We may have an intuition of what $Sem[\![X]\!]$ is for $X \in DK$. The state transitions are abstracted into the meaning of the object (named X) containing the distributed knowledge system Σ, on condition that there is some model (I_k, J) for each R_k within the distributed system Σ. Now let

$$\begin{array}{ll} Sem[\![(s)null]\!] = \Omega & (s \in S) \\ Sem[\![\lambda]\!] = \Lambda & (\lambda \in DK^*) \\ Sem[\![x]\!] = [\![x]\!] & (x \in DK^* - \{\lambda\}) \end{array}$$

With these notations, let $\mathcal{A}$ be inductively defined in Backus Normal Form:

$$\mathcal{A} ::= \Omega \mid \Lambda \mid [\![x]\!] \mid \mathcal{A} + \mathcal{A} \mid \mathcal{A} \circ \mathcal{A}$$

Definition 5. For α, $\beta \in \mathcal{A}$, we define $\alpha = \beta$, if $\alpha(s) = \beta(s)$ for any $s \in S$.

(1) The addition $+$ is defined on $\mathcal{A}$: For $s \in S$,

$$(\alpha + \beta)(s) = \alpha(s) \cup \beta(s)$$

(2) The multiplication $\circ$ is defined on $\mathcal{A}$: For $s \in S$,

$$(\alpha \circ \beta)(s) = \cup_{t \in \alpha(s)} \beta(t)$$

We have seen that $\Omega(s) = \emptyset$ and $\Lambda(s) = \{s\}$ for any $s \in S$. It is seen by induction that $[\![x]\!] \circ [\![y]\!] = [\![xy]\!]$ for $x, y \in DK^*$. Without such reduction to $[\![xy]\!]$ from $[\![x]\!] \circ [\![y]\!]$, we pay attention to the algebraic structure of $\mathcal{A}$. Then we can have a semiring $(\mathcal{A}, +, \circ, \Omega, \Lambda)$.

6 CONCLUSION

The algebraic structure of references to website with situations is abstractly formulated by means of recursion of reference constructions, to make complex structures of references clearer. The algebraic structure is equipped with fixed point model, when the mapping, associated with the recursively constructed references, may have a consistent fixed point such that rejective and suspended negations cannot be contradictory to the acquisitive positive..

About algebraic structure of a distributed knowledge system, organized by a tuple of algebraic structures as multiple knowledge bases, and sequential applications of such systems, we have the results, in knowledge management technologies: (a) A state constraint structure is formulated such that it may represent a distributed knowledge system and be modeled by fixed point theory. (b) The structure contains common negatives among knowledge bases constituting the whole knowledge system. (c) As regards sequential application of the distributed knowledge systems, the traverses of states contain algebraic semiring, with multiplication for the concatenation (to form sequences) and with addition for nondeterministic choices of state transitions, which are caused by consecutive applications of distributed knowledge systems.

REFERENCES

Bentham, J. V., Mierzewski, K., and Blando, F. (2022). The modal logic of stepwise removal. *Rev.Symb.Log.*, 15(1):36–63.

Bezhanishvili, N., G.Grilletti, and Quadrellaro, D. (2022). An algebraic approach to inquisitive and dna-logics. *Rev.Symb.Log.*, 15(4):950–990.

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.

Egre, P., Marty, P., and Renne, B. (2021). Knowledge, justification and adequate reasoning. *Rev.Symb.Log.*, 14(3):681–727.

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

Gebser, M. and Schaub, T. (2016). Modeling and language extensions. *AI Magazine*, 3(3):33–44.

Gessel, T. V. (2022). Questions in two-dimensional logic. *Rev.Log.Comput.*, 15(4):859–879.

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

Kaufmann, B., Leone, N., Perri, S., and Schaub, T. (2016). Grounding and solving in answer set programming. *AI Magazine*, 3(3):25–32.

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

Pawlowski, P. and Urbaniak, R. (2018). Many-valued logic of informal provability: A non-deterministic strategy. *Review.Symb.Log.*, 11(2):207–223.

Rasga, J., Sernadas, C., and Carnielli, W. (2021). Reduction techniques for proving decidability in logics and their meet-combination. *Bull.Symb.Log.*, 27(1):39–66.

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

Rin, B. and Walsh, S. (2016). Realizability semantics for quantified modal logic: generalizing flagg's 1985 construction. *Rev.Symb.Log.*, 9(4):752–809.

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

Tennant, N. (2021). What is a rule of inference. *Rev.Symb.Log.*, 14(2):307–346.

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

Yamasaki, S. and Sasakura, M. (2021). Algebraic expressions with state constraints for causal relations and data semantics. In *Communications in Computer and Information Science 1446*, pages 245–266.

Yamasaki, S. and Sasakura, M. (2023). Abstraction of prevention conceived in distributed knowledge base. In *Proceedings of the 6th International Conference on Complexity, Future Information Systems and Risk*, pages 39–46.

Yamasaki, S. and Yokono, H. (2008). An analytic method of seasonal reference in japanese haiku-poem. *IPSI Trans. on AR*, 4(1):27–33.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS