Logical Rule Set to Data Acquisition and Database Semantics

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: Logic, Data Acquisition, Database Semantics.

Abstract: This paper is concerned with website page references including streams from on-line seminar, as reference data of website organization. The motive to take data processing of website pages comes from our observation that the website page references contain a structure of data acquisition and of logical database framework. As formality of data processing, we then treat logical expressions in intuitionistic propositional logic. By evaluating linkage of website pages as well as balked and suspended negatives of link, we make analysis of the structure contained in processes of using website page references, for the purpose of data acquisition and database semantics. As logical expressions, we make use of logical rules, from the views of structural analysis of linkage consistency. By means of query derivation to the logical rule set, we can have 3-valued domain model theory in logical rule sets. The query derivation of this paper is a newly designed method for data acquisition. With respect to abstraction of the state notion from computing environments, a logical database is formulated as a state constraint rule set with data acquisition capability, causing state transitions. The behavioural meaning of logical databases is captured in modal operator such that we can apply a modal logic to meaning descriptions of logical databases. With a different level of logical framework for the meaning of database, modal logic is presented. Apart from the level of logical database in intuitionistic propositional logic, modal operator may be related to semantics for database. By means of fixed point of some function denoting a relation between states for the modal operator, computing-environment states are specified to be concerned with the logical databases.

1 INTRODUCTION

This paper aims at data acquisition and database semantics, in a logical framework with some query derivation, motivated by analysis of and by abstraction from data processing of website usages. We deal with processing based on website searches, where we pay attention to the structural consistency of a website page containing (i) references to the page and (ii) streams constructed by on-line seminar class, embedded in the page. The structure of page references is organized as a logical rule set such that structural consistency may be evaluated with a 3-valued domain. The streams organized as image and audio sequences are supposedly uploaded to some website page, where a reason by stream to reference is likely adequacy, based on the effect of stream. Rather than investigations of adequacy, the linked or the balked reference with an effect of stream is assumed in this paper. Then linkage of references is the primary point for data processing of this paper, whose consistency may be made clearer from logical viewpoints. As to linkage consistency, why a 3-valued domain is taken comes from the treatments: Firstly, the reference page may be linked, or suspended for some cause. Secondary, the stream effect is regarded as bringing the linked or balked reference, by means of conceived adequacy of stream reason. With such aims, two kinds of negation, for the balked and the suspended, are required with the positive to denote the linked status of references. The logical value from some 3-valued domain is to be assigned to the situation of rules with respect to consistency for linkage of pages.

For data processing concerning website searches, we would have a method of query derivation to examine its logical structure, which reflects logical reasoning of new type. For abstraction from processing to data acquisition, studies of logic for knowledge are of use. In logic for processing, our approach is correlated to the backgrounds:

[a] https://orcid.org/0000-0001-7895-5040
[b] https://orcid.org/0000-0002-8909-9072

Yamasaki, S. and Sasakura, M.
Logical Rule Set to Data Acquisition and Database Semantics.
DOI: 10.5220/0012710000003756

In *Proceedings of the 13th International Conference on Data Science, Technology and Applications (DATA 2024)*, pages 237-244
ISBN: 978-989-758-707-8; ISSN: 2184-285X

(a) Nonmonotonic logic on reasoning and inference mechanism should be applicable to design and programming (Reiter, 2001). This paper examines data acquisition which contains nonmonotonic reasoning aspects, with query derivation as actions. As dynamic aspects of knowledge processing, reduction of decidability (Rasga et al., 2021), reduction in rewriting calculus (Bertolissi et al., 2006) and inference theory (Tennant, 2021) are remarked, as well as dynamic logic for action (Spalazzi and Traverso, 2000). This paper will take modal logic with respect to relations between states, possibly caused by query derivation regarding database, rather than actions as in dynamic logic.
(b) Modal logic based on state space in Kleene-Kripke theory should be taken into considerations, to present database semantics. Modal mu-calculus may be applicable to the case of this paper. As backgrounds, not only propositional logic but also first-order modal logic is formally dealt with (Fitting, 2002). And then, there have been developments from several viewpoints. Second-order abstraction is established for modality. Ambiguity of knowledge may be discussed (Kooi, 2016), as well as quantifiers over epistemic agents (Naumov and Tao, 2019). For knowledge processing and computability, quantified modal logic is discussed (Rin and Walsh, 2016). Topological space is presented, as studies for modality (Goldblatt and Hodkinson, 2020) and dynamic modality (Bentham et al., 2022).

This paper is concerned with data processing which is to primarily detect consistency in linkage of website page references including stream adequacies to those references. Thus reference structure should be logically analyzed such that nonmonotonic logic from the backgrounds may be most related to the present problem. We see that the logical structure, as a rule set, can be analyzed with query derivation. The derivation is refined from negation by failure rule in nonmonotonic logic such that it may work in the logical structure of this paper, under both possibility of positive and negative valuations. The derivation of this paper is to be a method for data acquisition, where acquisition is abstracted from detection of the proposition evaluated as truth. In terms of logical structure, state constraint database is next formulated. Semantics should be the task to be examined, in our setting of logical structure to database.

Because databases are settled in a distributed system, process with communication may be relevant to formulations of database semantics. For the traverses of states in distributed processes, we note concepts on state transitions and their effects, from the relevant backgrounds: Sequences traversing states in a distributed system may be closely related to the method of automata (Droste et al., 2009). Its framework may be adopted, for the abstract state machine which the database of this paper can be formulated by.

We interpret the state constraint database of this paper as causing state transitions. Based on such a formulation of database classes, modal operator conceiving database class are definable to present relations on a state space. The modal operator is to be embedded into Hennnessy-Milner logic, as well. By such logic, a fixed point semantics can be considered for the representation of states, including databases causing state transitions.

2 REFERENCE DATA

We make analysis on data processing with human computer interaction (HCI). With an experience of having constructed on-line seminar class by virtual reality methods, we may use stream of image and audio (as HCI), as uploaded to the website. At the same time, the website page can in general contain page references, by which we have access to other website pages. On-line seminar class to website demonstration is in stream data. The structure of website page references with streams is logically captured, aiming at linkage relations on websites, such that both streams and page reference may be the objects of reference data.

Logical Rule Set
The logical expression, which we here deal with, contains the linked, the balked and the suspended references, where the balked reference comes from the status of stream data (which the reference is interpreted as denoting), linked or balked.

We have constructed a version of virtual reality seminar class as HCI, which may implement several functions of real-time, on-line seminar class in a stream: (i) Images taken by video cameras may be organized, with audio, into a virtual reality class (VR-class) of streams, where attenders can take part in the class by the computing facility. (ii) The images can include not only a presenter with blackboards but also class attenders. (iii) The class can include any angle from the point of video camera shots. Several blackboards in a class may be observed. (iv) Some zoom up to the blackboard is implemented, for the attenders to observe the contents

Such HCI for attenders is organized as streams of sensed image and audio having caught seminar presentation. The streams may be uploaded to a website. Then data processing can be implemented, with uploaded streams, as well as with page reference queries, to make use of website searches: (a) Seminar presentation is sensed as image and audio forming streams. (b) Streams are uploaded to website pages from which attenders make queries, where streams are viewable by attenders.

The page reference is now interpreted as a logical proposition, with intention to abstract logical database from propositional data regarding linkage of references. The seminar streams may be organized by the method of phrase structure grammar like an interactive constraint system with leftmost derivations (Yamasaki, 2007). In this paper, the seminar class in stream *st* to website reference is supposedly expressible as (i) $\langle st\rangle p$ with a modal operator $\langle st\rangle$ and a proposition p, or (ii) a reason, *st* (Egre et al., 2021), where a form of $st : p$ (with *st* supporting a proposition p) may be taken as a reference. As a stage of processing, instead of modality, stream *st* is here treated as adequate reason for p to be linked or balked. Except streams, the other page references are linked or suspended. So far a website page reference may be recursively constructed as in Table 1, where the page reference is presented by a proposition without negation (as the linked) or with balked or suspended negation. The balked negation is expressed by *not*, while the suspended negation is expressible by default $\sim$.

The website page is here regarded structurally as a logical rule and the website pages are as in a set of logical rules, where the balked or the suspended references are represented as propositions with negations, and the linked reference is a proposition without negation.

By Backus-Naur Form, we then define a finite or countably infinite set F of rules with propositions in an assumed set.

$$F ::= \emptyset \mid Impli \cup F$$
$$Impli ::= Prem \Rightarrow Conclu$$
$$Prem ::= \emptyset \mid \{p\} \cup Prem \mid \{not\ p\} \cup Prem \mid \{\sim p\} \cup Prem$$
$$Conclu ::= p \mid \sim p$$

where (i) $\emptyset$ denotes the empty set, as the rule set F and as the set *Prem*, (ii) p is a variable ranging over an assumed set P of propositions, and (iii) the arrow $\Rightarrow$

Table 1: A recursive structure is captured as a rule.

Page p	Containing
Linked page references	$p_{link_1}, \ldots, p_{link_n}$ ...
Page references with Streams (Linked or Balked)	$p_{st_1}, \ldots, p_{st_m}$ ...
Suspended page references	$p_{sus_1}, \ldots, p_{sus_l}$ ...

corresponds to the implication, in the sense as below defined.

Note that *Prem*, *Conclu*, and *Impli* are to express premise, conclusion, implication, respectively, for a rule (with variable p in P). *Conclu* is not concerned with stream reference so that it may contain a reference, to be linked or suspended.

Evaluation of Rule Sets

What the proposition denotes is defined, reflecting linkages of page references and consistency of page references, such that the proposition can be considered as abstraction of page (reference) and may form logical rule sets. With two kinds of negations for the balked and the suspended (based on intuitionistic propositional logic), we adopt a 3-valued domain *Dom*, including "the unknown" in addition to truth and falsity, and make use of a bounded lattice *Dom* $= (\{f, unk, t\}, \bigvee, \bigwedge, \bot, \top)$ equipped with the partial order $\sqsubseteq$ (by which $f \sqsubseteq unk \sqsubseteq t$), and an implication $\Rightarrow$ may be taken:

(a) $\bot = f$ and $\top = t$ are the least and the greatest elements of the algebra (set) $Dom_v = \{f, unk, f\}$, respectively, with respect to the partial order $\sqsubseteq$.

(b) The least upper bound (*join*, with $\bigvee$) and the greatest lower bound (*meet*, with $\bigwedge$) exist for any two elements of $\{f, unk, t\}$.

(c) The implication (with $\Rightarrow$) is defined in a way that $z \sqsubseteq (x \Rightarrow y)$ iff $x \bigwedge z \sqsubseteq y$.

With the bounded lattice *Dom*, we have a valuation $V : P \rightarrow Dom_v$. With respect to V, we have the value $val_V(E)$ of the expression E for *Prem*, *Conclu*, *Impli* and F, recursively defined as follows.

(1) For p, $not\ p$ and $\sim p$:

$$val_V(p) = V(p)\ (p \in P)$$
$$val_V(not\ p)$$
$$= \text{if } (val_V(p) = f) \text{ then } t \text{ else } f$$
$$val_V(\sim p)$$
$$= \text{if } (val_V(p) = f) \text{ then } t$$
$$\text{else if } (val_V(p) = unk) \text{ then } unk \text{ else } f$$

(2) For *Prem*, *Conclu* and *Impli*:

$val_V(Prem)$
$= \text{if } \forall x \in Prem.(val_V(x) = t) \text{ then } t$
else if $\exists y \in Prem.(val_V(y) = f)$ then f else unk
(x, y are variables ranging over the set *Prem*, respectively, such that $val_V(\emptyset) = t$ for $Prem = \emptyset$.)
$val_V(Conclu)$
$= \text{if } (Conclu = p) \text{ then } val_V(p)$
else if $(Conclu = \sim p)$ then $val_V(\sim p)$
$val_V(Impli) = val_V(Prem \Rightarrow Conclu)$
$= \text{if } val_V(Prem) \sqsubseteq val_V(Conclu) \text{ then } t$
else if $(val_V(Conclu) = f)$ then f else unk

(3) For the rule set F:

$val_V(F)$
$= \bigwedge_{Imp \in F}\ val_V(Impli)$
$(val_V(\emptyset) = t$ and $\bigwedge$ operates on the countable set F.)

Note the treatment of a logical expression $st : p$ denoting a reference (as a proposition p) with an effect of a stream st. With respect to the effect of st in terms of adequacy, the linkage status of the reference as p is evaluated:

$$val_V(st : p) = \begin{cases} val_V(p) & (st \text{ is adequate and } p \text{ is linked}) \\ val(not\ p) & (st \text{ is inadequate and } p \text{ is balked}) \end{cases}$$

This paper is concerned with propositional logic for logical database abstraction, such that the expression $st : p$ may be reduced to p or $not\ p$ within each rule.

Example 1: Assume the rule set (where the rule is delimited by the semicolon), even with a stream st, where $demanded$, $examined$, $specified$ and $supplied$ are propositions.

$\emptyset \Rightarrow demanded;$
$\{st : examined, demanded\} \Rightarrow supplied;$
$\{\sim examined\} \Rightarrow \sim supplied;$
$\{\sim specified\} \Rightarrow examined$

In Section 3, we have two rule sets, depending on whether $st : examined$ is, with effect of st, reduced to *examined* or *not examined*.

3 LOGICAL RULE SET AND DATA ACQUISITION

Towards logical database framework, query derivation is studied with respect to model theory in the logical rule sets. Within a formulated logical set of rules, query derivation is well defined, as data acquisition.

3.1 Query to Logical Rule Set

The assignment of $V : P \to Dom_v$ (as in Section 2) is now specified to a pair $(I,J) \in 2^P \times 2^P$. For the pair (I,J) such that $I \cap J = \emptyset$, i.e., I and J are disjoint, the valuation $V(I,J) : P \to \{f, unk, t\}$ is defined to be:

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

The value $val_{V(I,J)}(F)$ may be settled inductively, as defined in Section 2. If the disjoint pair (I,J) $(I \cap J = \emptyset)$ causes $val_{V(I,J)}(F) = t$, the pair is called a model of the rule set F.

Query Derivation

Let G be a variable ranging over the set *Prem*.
(a) $suc(G)$ means that G succeeds in the derivation.
(b) $fail(G)$ stands for G's failure of the derivation.

Assume a rule set F over the set P of propositions. We have the recursively defined derivations with a variable G' over *Prem*, as follows:

(1) If $G = \emptyset$, then $suc(G)$.

(2) If $G = \{p\} \cup G'$ and
$\exists(Prem_1 \Rightarrow p) \in F.$
$((\forall(Prem_2 \Rightarrow \sim p) \in F.\ fail(Prem_2))$
and $(suc(Prem_1 \cup G')))$,
then $suc(G)$.

(3) If $G = \{not\ p\} \cup G'$ such that $fail(\{p\})$ and $suc(G')$, then $suc(G)$.

(4) If $G = \{\sim p\} \cup G'$ such that $fail(\{p\})$, and $suc(G')$, then $suc(G)$.

(5) If, for p in F, $\forall(Prem \Rightarrow p) \in F.\ fail(Prem)$, then $fail(\{p\})$.

(6) If $fail(\{p\})$, then $fail(\{p\} \cup G)$.

(7) If $suc(\{p\})$, then $fail(\{not\ p\})$.

(8) If $suc(\{p\})$, then $fail(\{\sim p\})$.

Example 2: Following the rule set F as in Example 1, we have rule sets F_1 and F_2.

F_1 : $\emptyset \Rightarrow demanded;$
$\{examined, demanded\} \Rightarrow supplied;$
$\{\sim examined\} \Rightarrow \sim supplied;$
$\{\sim specified\} \Rightarrow examined$

F_2 : $\emptyset \Rightarrow demanded;$
$\{not\ examined, demanded\} \Rightarrow supplied;$
$\{\sim examined\} \Rightarrow \sim supplied;$
$\{\sim specified\} \Rightarrow examined$

From F_1,
$suc(\{demanded\})$, $suc(\{examined\})$,
$suc(\{supplied\})$,
$fail(\{specified\})$
causing $suc(\{\sim specified\})$ and thus
$suc(\{examined\})$
(with the rule: $\{\sim specified\} \Rightarrow examined$).

From F_2,
$suc(\{demanded\})$, $suc(\{examined\})$,
$fail(\{specified\})$, $fail(\{supplied\})$.

Regular Model
We have a specified model of a given rule set, which succeeding and failing derivations for query may be sound with respect to.

Definition 1. Let a model (I,J) of a rule set F (over the proposition set P) satisfy the condition:
$\forall q \in P$. (if $\forall(Prem \Rightarrow q) \in F$. $(val_{V(I,J)}(Prem) = f)$, then $q \in J$).
The model (I,J), satisfying the condition, is called a regular model of F.

Proposition 2. *Assume that the pair (I,J) (in $2^P \times 2^P$) is a regular model of a rule set F (over the proposition set P). With query derivation,*

(a) *If $suc(\{p\})$, then $p \in I$.*
(b) *If $fail(\{q\})$, then $q \in J$.*

Proof. (i) In case that $suc(\{p\})$, then, by the construction of the derivation,
(a) $\exists(Prem \Rightarrow\ p) \in F.\ suc(Prem)$.
(b) $\forall(Prem' \Rightarrow \sim p) \in F.\ fail(Prem')$.

As regards (a), if $Prem = \emptyset$, then $val_{V(I,J)}(p) = t$ (i.e., $val_{V(I,J)}(\emptyset) = t$ and $val_{V(I,J)}(Prem \Rightarrow p) = t$, for (I,J) to be a model of F). Thus $p \in I$.
If $Prem \neq \emptyset$, then $\forall x \in Prem.\ suc(\{x\})$ such that:
$suc(\{q_1\})$, or
$suc(\{not\ q_2\})$ (from $fail(\{q_2\})$), or
$suc(\{\sim q_3\})$ (from $fail(\{q_3\})$),
respectively, for $x = q_1$, $not\ q_2$, or $\sim q_3$. By applying induction,
$q_1 \in I$, or $q_2 \in J$, or $q_3 \in J$.
It follows that
$val_{V(I,J)}(Prem) = t$, and thus $val_{V(I,J)}(p) = t$
for $val_{V(I,J)}(Prem \Rightarrow p) = t$ to follow the model (I,J). Thus $p \in I$.

As regards (b), $\exists y \in Prem'.\ fail(\{y\})$.
If $y = r_1$, $not\ r_2$, or $\sim r_3$, then $fail(\{r_1\})$, $suc(\{r_2\})$ (from $fail(\{not\ r_2\})$), or $suc(\{r_3\})$ (from $fail(\{\sim r_3\})$), respectively, which, by applying induction, means that
$r_1 \in J$, $r_2 \in I$, or $r_3 \in I$.
It is thus reasoned that $val_{V(I,J)}(Prem') = f$. Even with $p \in I$, $val_{V(I,J)}(Prem' \Rightarrow \sim p) = t$, which is consistent for (I,J) to be a model of F.
(ii) In case that $fail(\{q\})$, then, by the construction of the derivation,
$\forall(Prem_1 \Rightarrow q) \in F.\ fail(Prem_1)$.
(a) If there is no rule of the form $Prem_1 \Rightarrow q$, then $q \in J$, because of the regular model condition.
(b) For any rule of the form $Prem_1 \Rightarrow q$, assume that $fail(Prem_1)$ (caused by $fail(\{q\})$). By the same induction for (b) in case of (i), we see that $val_{V(I,J)}(Prem_1) = f$. By the condition of the regular model (I,J), $q \in J$.

This is consistent to the regular model (I,J) with respect to any rule of the form $Prem_2 \Rightarrow \sim q$, because $val_{V(I,J)}(\sim q) = t$ and thus
$val_{V(I,J)}(Prem_2 \Rightarrow \sim q) = t$. □

We therefore see that the query derivation is sound with respect to the regular model, when there is a regular model of the given logical rule set. Although it may be in general, for an infinite logical rule set, unknown whether or not there is a regular model of it, we regard Proposition 2 as significant for data acquisition based on query to the given logical rule set.

3.2 Model Theory in Logical Rule Set

We study a model related to query derivation. In Proposition 3 as follows, some derivation for query is of use to have a model of a given logical rule set.

Since the derivation does not contain any routine for the balked negation but for the suspended negation, we discuss the case for a logical rule set, where the suspended negation replaces the balked negation, as well.

Proposition 3. *Given a rule set F over the set P of propositions, let*

$$I = \{p \in P \mid suc(\{p\})\},\ and$$
$$J = \{q \in P \mid fail(\{q\})\}$$

with the derivation where the conditions as follows are assumed.

(a) $I \cap J = \emptyset$.
(b) $\forall r \notin (I \cup J)$. $((\forall(Prem \Rightarrow r) \in F$.
(it is not the case that $suc(Prem)))$ *and*
$(\forall(Prem' \Rightarrow \sim r) \in F$.
(it is not the case that $suc(Prem'))))$.

Then (I,J) is a regular model of F.

Proof. (1) For the pair (I,J) to be a model of F, it is proved by induction on the structure of the rule set F, with the assumptions. For any proposition r occurring in F such that $r \in I, J$, or $r \notin (I \cup J)$, we examine each case of the rules in F:
(i) In case that $r \in I$, we see that with $suc(\{r\})$ and $val_{V(I,J)}(r) = t$,

$$\forall(Prem \Rightarrow r) \in F.\ (val_{V(I,J)}(Prem \Rightarrow r) = t).$$

At the same time,
$\forall(Prem' \Rightarrow \sim r) \in F.\ fail(Prem')$.
(If there is no rule of the form $Prem' \Rightarrow \sim r$, the following examination is not needed.)
It follows by induction on query derivation that
$\exists r_1 \in Prem'.\ fail(\{r_1\})$ $(r_1 \in J)$ or
$\exists not\ r_2 \in Prem'.\ fail(\{not\ r_2\})$
(i.e., suc$(\{r_2\})$ and $r_2 \in I$) or
$\exists \sim r_3 \in Prem'.\ fail(\{\sim r_3\})$
(i.e., $suc(\{r_3\})$ and $r_3 \in I$).
By $r_1 \in J$ or $r_2 \in I$ or $r_3 \in I$, we have
$val_{V(I,J)}(Prem') = f$ (to $val_{V(I,J)}(\sim r) = f$),
such that $val_{V(I,J)}(Prem' \Rightarrow \sim r) = t$.

(ii) In case that $r \in J$, we see that with $fail(\{r\})$ and $val_{V(I,J)}(r) = f$, $\forall(Perm_1 \Rightarrow r) \in F.\ fail(Perm_1)$.
Then there is no rule of the form $Prem_1 \Rightarrow r$ in F, or we can have
$val_{V(I,J)}(Perm_1) = f$,
for the same reason as in (i) that $fail(Perm')$ causes $val_{V(I,J)}(Perm') = f$. Thus $val_{V(I,J)}(Perm_1 \Rightarrow r) = t$.
At the same time, with $val_{V(I,J)}(\sim r) = t$,
$\forall(Perm_2 \Rightarrow \sim r) \in F$.
$(val_{V(I,J)}(Perm_2 \Rightarrow \sim r) = t)$.
(Even if there is no rule of the form $Prem_2 \Rightarrow \sim r$, the examination is consistent.)
(iii) In case that $r \notin (I \cup J)$, by the assumption of the model (I,J), a rule of the form $Prem \Rightarrow r$ is in F, and

$\forall(Prem \Rightarrow r) \in F$.
(it is not the case that $suc(Prem)$).

Then, by the construction of query derivation,
$\exists x \in Prem.$(it is not the case that $suc(\{x\})$).
If $x = r_1$ ($r_1 \in P$), then we have not got $suc(\{r_1\})$. If $x = not\ r_2$ ($r_2 \in P$), then we have not $suc(\{not\ r_2\})$, nor $fail(\{r_2\})$. If $x = \sim r_3$ ($r_3 \in P$), then neither $suc(\{\sim r_3\})$ nor $fail(\{r_3\})$. When $x \in Prem$ is r_1, $not\ r_2$ or $\sim r_3$, $r_1 \notin I$ or $r_2 \notin J$ or $r_3 \notin J$, respectively. Therefore, $val_{V(I,J)}(Prem) \neq t$, such that $val_{V(I,J)}(Prem \Rightarrow r) = t$.
By the assumption of the pair (I,J),

$\forall(Prem' \Rightarrow \sim r) \in F$.
((it is not the case that $suc(Prem')$).

(If there is no rule of the form $Prem' \Rightarrow \sim r$, the following examination is not needed.)
For the same reason (as above) to cause
$val_{V(I,J)}(Prem) \neq t$
from no case of $suc(Perm)$, we also have
$val_{V(I,J)}(Prem') \neq t$.
With $r \notin (I \cup J)$ such that $val_{V(I,J)}(r) = unk$ and $val_{V(I,J)}(\sim r) = unk$, we may conclude that
$val_{V(I,J)}(Prem' \Rightarrow \sim r) = t$.
(2) With the pair (I,J) shown to be a model of F in (1), assume for any $q \in P$ that

$$\forall(Prem \Rightarrow q) \in F.\ (val_{V(I,J)}(Prem) = f).$$

It follows from the construction (i.e., definition) of the model (I,J) that $fail(Prem)$. By the construction of the derivation, we have $fail(\{q\})$. That is, $q \in J$. Thus the model (I,J) is a regular model. □

Example 3: To the rule sets F_1 and F_2, we have the following two sets F_3 and F_4 (which are obtained by extensions with proposition *timely* included in F_1 and F_2, respectively). We then examine whether or not there may be regular models of the rule set F_3 and F_4.

F_3 : $\emptyset \Rightarrow demanded$;
$\{examined, demanded\} \Rightarrow supplied$;
$\{\sim examined\} \Rightarrow \sim supplied$;
$\{\sim specified\} \Rightarrow examined$;
$\{\sim specified\} \Rightarrow timely$;
$\{\sim timely\} \Rightarrow specified$

F_4 : $\emptyset \Rightarrow demanded$;
$\{not\ examined, demanded\} \Rightarrow supplied$;
$\{\sim examined\} \Rightarrow \sim supplied$;
$\{\sim specified\} \Rightarrow examined$;
$\{\sim specified\} \Rightarrow timely$;
$\{\sim timely\} \Rightarrow specified$

For both F_3 and F_4,
$(\{demanded \mid suc(\{demanded\})\}, \emptyset)$
can be a model in the sense of Proposition 3. Note that it is a regular model for both F_3 and F_4.

Replacement of Balked Negation by Suspended Negation
We have some properties in a rule set F without balked negation, where the proofs of the propositions might be shown.

Definition 4. Given a rule set F (over the proposition set P), let

$Prem[\sim/not]$
$= \{p \mid p \in Prem\} \cup \{\sim q \mid not\ q \in Prem\}$
$\cup \{\sim r \mid \sim r \in Prem\}$, and
$F[\sim/not]$
$= \{Prem[\sim/not] \Rightarrow Conclu \mid Prem \Rightarrow Conclu$ in $F\}$.

Proposition 5. *Given a rule set F, assume that the pair (I,J) is a model of the rule set $F[\sim/not]$. Then (I,J) can be a model of F.*

Definition 6. Given a rule set F (over the proposition set P), we say that F satisfies *suspension-condition*, if *Conclu* of any rule $Prem \Rightarrow Conclu$ in F is free from the occurrence of the negation $\sim$.

Proposition 7. *Assume that a rule set F satisfies suspension-condition. Then there is a model of F.*

4 DATABASE SEMANTICS WITH ABSTRACT STATES

Logical database with state constraint rule set is formulated such that the logical databases may conceive abstract state machine causing state transitions by the effects of (regular) models of rule sets.

State Transitions with Regular Model
A rule set is regarded as causing state transitions, with query derivations. The database (with a constraint state) D is to denote the set of databases of the form

$(s)F$ (with a state s constraining the rule set F of Sections 2). D^* stands for the set of finite sequences from D including the empty sequence λ. The sequence of D^* may denote a finite number of sequential applications of databases in D. The semantics for the sequence in D^* is defined inductively as follows, for a state set S and its power set 2^S. If there is a regular model of F, F may make the assignment of a state set, to the constraint state of F.

The function $sem : D^* \to (S \to 2^S)$ is defined for a state $s \in S$ as follows:

$$sem[\![(s')null]\!](s) = \emptyset \text{ (with any } s' \in S)$$
$$sem[\![\lambda]\!](s) = \{s\}$$
$$sem[\![(s')F]\!](s) = \begin{cases} S_1 & \text{if } s = s' \text{ and there is a regular model of } F \\ & \text{such that } F \text{ assigns } S_1 (\subseteq S) \\ \emptyset & \text{otherwise} \end{cases}$$

and recursively

$$sem[\![Uv]\!](s) = \cup_{t \in sem[\![U]\!](s)}\, sem[\![v]\!](t) \ (U \in D, v \in D^*)$$

The denoted set $sem[\![(s')null]\!](s)$ (in 2^S) is uniquely defined as $\emptyset$ for any s'. We may have an intuition of what $sem[\![U]\!]$ is for $U \in D$. The state transitions are abstracted into the meaning of the database U (containing a rule set F), on condition that there is some regular model (I,J) for F.

Application of Modal Logic

The modal mu-calculus contains a fixed point notation to reflect abstract states of environments where actions and communications are satisfactory for given conditions. In the context of programming system formulations, Hennessy-Milner Logic was presented, in relation to concurrency by M.Hennessy et al. (1985).

As a meta-expression level of logic, we make use of modal logic and present meaning of logical database of the above described in intuitionistic propositional logic. The modal logic, which we present as a meta-expression for the database constrained by a state, may be of use, with fixed point operator as in systems (Venema, 2006; Venema, 2008).

Logical Database in Modal Operator

The set Φ of (logical) formulas concerning with a state set are defined inductively with postfix modality.

$$\varphi ::= \mathrm{ff} \mid \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi\langle d\rangle$$

The intuitive meanings of symbols are described as below, where the formal meanings are given with the transition system: (a) ff is the falsity. (b) φ denotes proposition in Φ. (c) $\neg$ is the logical negation, and $\vee$ stands for the disjunction. (d) $\langle d\rangle$ is a postfix modal operator concerning database set d in 2^D (with the database set D).

The proposition in the set Φ is not common with the proposition contained in the database set D.

A Transition System $\mathcal{T}$:

For the set Φ of formulas, a transition system $\mathcal{T}$ is defined to be a quadruplet (S, D, Re, Val) where: (i) S is a set of states. (ii) D is a set of databases. (iii) Re maps to each $d \subseteq D$ a relation $Re(d)$ on S. (iv) Val maps to each proposition φ a subset of S (a state set in 2^S).

The above state set S might be common to the state set regarding the semantic function sem.

Meaning of Formulas

The meaning of a formula may be a subset of the state set (in the transition system), which follows the way of Hennessy Milner Logic (HM-Logic).

Given a transition system $\mathcal{T}$, the functions $[\![\cdot]\!]_{HM}$ are defined as meanings of formulas.

(a) $[\![\mathrm{ff}]\!]_{HM} = \emptyset$ (in 2^S).

(b) $[\![\varphi]\!]_{HM} = Val(\varphi)$.

(c) $[\![\neg\varphi]\!]_{HM} = S - [\![\varphi]\!]_{HM}$, and $[\![\varphi_1 \vee \varphi_2]\!]_{HM} = [\![\varphi_1]\!]_{HM} \cup [\![\varphi_2]\!]_{HM}$.

(d) $[\![\varphi\langle d\rangle]\!]_{HM} = \{s' \in S \mid \exists s \in [\![\varphi]\!]_{HM}.\ sRe(d)s'\}$, where $sRe(d)s'$ iff $s' \in sem[\![(s)F]\!]$ (with the semantic function sem) for some $(s)F \in d \subseteq D$.

We can have the formula $\varphi[d]$, by the representations:

$\mathrm{tt} = \neg\,\mathrm{ff}$, $\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2)$ and
$\varphi[d] = \neg((\neg\varphi)\langle d\rangle)$

Definition 8. With a given database set $d \subseteq D$ and a variable $[\![X]\!]_{HM}$ ranging over the set 2^S, let $g_d : 2^S \to 2^S$ be defined as $g_d([\![X]\!]_{HM}) = [\![X\langle d\rangle]\!]_{HM}$.

We easily see:

Proposition 9. *Given a database set $d \subseteq D$, if $[\![X]\!]_{HM} \subseteq [\![Y]\!]_{HM}$, then $g_d([\![X]\!]_{HM}) = [\![X\langle d\rangle]\!]_{HM} \subseteq [\![Y\langle d\rangle]\!]_{HM} = g_d([\![Y]\!]_{HM})$.*

As regards the function g_d for a given database set $d \in D$, we have a meaning of g_d by means of $[\![X]\!]_{HM}$ which satisfies $[\![X]\!]_{HM} = [\![X\langle d\rangle]\!]_{HM} = g_d([\![X]\!]_{HM})$ (a fixed point of g_d).

We can see the greatest fixed point of g_d with respect to subset inclusion in 2^S, in that g_d is monotonic (i.e., in the sense of Proposition 9). Following the classical result, we have:

Proposition 10. *Given a database set $d \subseteq D$, the greatest fixed point of g_d is given as*
$\cup\{[\![X]\!]_{HM} \mid [\![X]\!]_{HM} \subseteq [\![X\langle d\rangle]\!]_{HM}\}$,
with $g_d([\![X]\!]_{HM}) = [\![X\langle d\rangle]\!]_{HM}$.

The fixed point in modal logic contains the meaning consisting of states, such that it may abstractly represent environments conceived by states. The environments may contain data and operations in the sense of states. $[\![X\langle d\rangle]\!]_{HM}$ is an environmental state set after applying the set d (of databases) for the variable $[\![X]\!]_{HM}$ (concerned with states). $[\![X]\!]_{HM}$ is an environmental state set. Thus the fixed point of g_d may present the stable state set where both are the same.

If the relation $Re(d)$ is reflexive, $[\![X]\!]_{HM} \subseteq [\![X\langle d\rangle]\!]_{HM} = g_d([\![X]\!]_{HM})$. When d is allowed to include the empty sequence $\lambda \in D^*$, the relation $Re(d)$ may be reflexive. It is well known that if $[\![X]\!]_{HM} \subseteq [\![X\langle d\rangle]\!]_{HM} = g_d([\![X]\!]_{HM})$, then $[\![X]\!]_{HM} \subseteq \nu(g_d)$, where $\nu(g_d)$ denotes the greatest fixed point of g_d. Therefore if $Re(d)$ is reflexive, then $[\![X]\!]_{HM} \subseteq \nu(g_d)$. This is partly the reason why the greatest fixed point of g_d is here remarked, rather than the least fixed point of g_d, where g_d is associated with the postfix modal operator $\langle d\rangle$ for a set d of databases.

5 CONCLUSION

We have got abstraction from reference data of website pages including the organization of streams of image and audio, for queries of pages. The structure of reference data may be captured as logical database. Primary results are summarized:

(a) With respect to linkage status of page references, we have formulated a structure of logical rule sets. The rule set is countable in intuitionistic propositional logic, where an infinite set may be regarded as in accordance to first-order logic with Herbrand base.

(b) We have got 3-valued domain model theory, based on query derivation, such that the presented logical rule set may contain (i) meaning of propositions able to be queried or not, as well as (ii) data acquisition, applicable to the cases of causal relation (Yamasaki and Sasakura, 2021), effectiveness (Yamasaki and Sasakura, 2022) and revised version from the knowledge base (Yamasaki and Sasakura, 2023).

(c) In terms of models for the logical rule set, the state constraint rule set is regarded as logical database causing transitions of abstract states. The transition of abstract states, caused by the class of logical databases of this paper, is associated with modal operator as a relation on the state space. The logical level of modal operator aims at meta-expressions of the object level for logical database in intuitionistic propositional logic. We see that the logical database can be embedded in Hennessy-Milner Logic with modal operator, expressible for the transformations of computing-environment states.

REFERENCES

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

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.

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.

Goldblatt, R. and Hodkinson, I. (2020). Strong completeness of modal logics over 0-dimensional metric spaces. *Rev.Log.Comput.*, 13(3):611–632.

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

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

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.

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

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

Yamasaki, S. (2007). An interactive constraint system and its relation to logic in AI. In *Proceedings of the 19th Belgium - Dutch Conference on Artificial Intelligence*, pages 299–306.

Yamasaki, S. and Sasakura, M. (2021). Algebraic expressions with state constraints for causal relations and data semantics. In *CCIS 1446, Data Management Technologies and Applications*, pages 245–266.

Yamasaki, S. and Sasakura, M. (2022). Consultation to effectiveness and logical meaning. In *Proceedings of the 7th International Conference on Complexity, Future Information Systems and Risk*, pages 57–64.

Yamasaki, S. and Sasakura, M. (2023). Abstraction of prevention conceived in distributed knowledge base. In *Proceedings of the 8th International Conference on Complexity, Future Information Systems and Risk*, pages 39–46.