Consultation to Effectiveness and Logical Meaning

Susumu Yamasaki[a] and Mariko Sasakura[b]
Department of Computer Science, Okayama University, Tsushima-Naka, Okayama, Japan

Keywords: Abstract Consultation, Causal Analysis, Abstract State Machine, Model Theory in Database.

Abstract: This paper deals with consultation as a role of consultant like a teacher in a virtual reality class for lessons. The consultation subject is taken in causal analysis and database design to conceive causal relation. The causal analysis is abstracted into effectiveness which is caused by requisites and prohibitions for effects. The database design is represented as recursive Backus-Naur Form for structural expressions. Then abstract consultation can be regarded as a function to transform effectiveness into recursive representation. The sense or meaning of such abstract consultation may be a semantics of conditioned formulas in modal logic. The modal logic is formulated in this paper, by extending postfix modality for name of database design as well as greatest fixed point operator for denoting an equilibrium state set where the transformation of effectiveness with prefix modality to design with postfix modality may be available. The conditioning of formulas is given by a state set so that the formula may be considered in terms of propositional base. As regards recursive representations in Backus-Naur Form for database design, model theory is established in 3-valued domain, for the sense of strong negation in accordance to prohibitions influencing effectiveness. Some fixed point is taken for the mapping associated with recursive representations, where the mapping is related to formations of models over 3-valued domain. The fixed point model can be seen as means of retrieval. However, because the mapping is generally a nonmonotonic function whose fixed point semantics is not always available, we have another method to adopt negation as failure for strong negation such that the model of a given recursive representation as database is always guaranteed if the representation takes a restricted form. Abstract consultation to effectiveness for database design is thus formulated, with model theory in database representations.

1 INTRODUCTION

Based on the views on cognitive managements, we try to deal with complexity of systems by simplifications to abstract formality. For expertise in human computer interaction, we here consider the function of consultant abstracted from a teacher in a virtual reality class of lessons, as consultation from causal analysis to design of database. As backgrounds to study abstract consultation for formality to simplify complexity in refined technologies, we have seen knowledge engineering refinements:
(i) Causal analysis with reasoning is fundamental, from knowledge engineering aspects (R. Reiter, 2001). It is objective knowledge for action that forms computational structure of designs to implement complex knowledgeable systems.
(ii) Nonmonotonic logic should be the subject for knowledge, even with respect to temporality (Hanks and McDermott, 1987).
(iii) Knowing is the subject to be formalized widely and rigidly (Kooi, 2016; Naumov and Tao, 2019).

The consultation is in this paper taken as a process. The idea to think of processes is related to abstract state machine and process algebra (R. Milner, 1999):
(i) Rewriting process with reductions for calculus is formulated in functionalism as in C. Bertolissi et al. (2006).
(ii) Mobile ambients are dealt with in process of behaviours (Cardelli and Gordon, 2000; Merro and Nardelli, 2005).
(iii) The compositions of transitions in automata are formulated in algebraic structures (Droste et al., 2009).

Following knowledge engineering, process algebra concepts and the idea on autonomy (Yamasaki and Sasakura, 2020), the purpose of this paper is to abstract the role of a consultant as a teacher of virtual reality class. The knowledge engineering views extract a consultation function from a causal analysis

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

Yamasaki, S. and Sasakura, M.
Consultation to Effectiveness and Logical Meaning.
DOI: 10.5220/0010977400003197
In *Proceedings of the 7th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2022)*, pages 57-64
ISBN: 978-989-758-565-4; ISSN: 2184-5034

of effects to recursive relation for design of database whose manipulations implement the effects conceived in causal analysis.

As regards effectiveness, computability or logical reasonability is the primary concept already established. In accordance with effects conceived in causal analysis, effectiveness is here captured as causal relations between requisites with prohibitions and effects. The design of database follows effectiveness, where recursive causal relation is transferred to the design and representation by Backus-Naur Form for database. The transition or transformation from causal analysis to design can be understood in abstract state machine or transition systems as interpretations for modal logic.

In modal logic, so many works are compiled. Among them, modal logic with fixed point operator (Dam and Gurov, 2002; Kozen, 1983; Venema, 2008) is significant for this paper, since the denotational semantics, that is, senses or meanings of consultation require fixed point operator. In this paper, for a nontrivial sense to be provided, the greatest fixed point is needed. The actions as behaviors are treated in modal logic (Giordano et al., 2000; Spalazzi and Traverso, 2000), where the consultation of this paper is an abstract action from behavioral senses, so that this work is relevant to modal logic achievements on actions. First-order modal logic (Fitting, 2002) and topological space in transition systems for modal logic (Goldblatt and Hodkinson, 2020) are to be kept in mind to the next step of advancements. In this paper for a simple outlook on, a primary step to denotation of abstract consultation is made in the propositional base, by means of postfix modality for design (action), and by fixed point (for causal analysis followed by design) as an equilibrium.

The paper is organized as follows. Section 2 is concerned with an example of the consultant in a virtual reality class, whose role would be an abstract consultation. The consultation involves causal relation analysis for design of database to implement causal relation. In Section 3, modal logic is prepared by taking postfix modality for representation of the design by name, and greatest fixed point for denotation of abstract consultation. Section 4 gives recursive representation of (database) design for causal analysis by Backus-Naur Form. The model theory of such representations are presented in 3-valued domain, for the effects caused by requisites and prohibitions. The model theory conceives nonmonotonic function treatments, where some approximate interpretation can be adopted for prohibited predicates of database. Concluding remarks are mentioned in Section 5.

2 ABSTRACT CONSULTATION

Paying attention to the consultant role of a teacher at a class to students, we abstract some consultation function for simplicity rather than for complex organization of refined implementation techniques. That is, simplicity by abstraction from effectiveness for analysis may be transformed to design of a system scheme. It is regarded as consultation function

- from analysis in problem solving, by effectiveness in causal relation,
- to (system) design of database, by recursive representation.

We have constructed a version of virtual reality class for lessons, which may implement a partial function of real-time online class for a teacher to be interactive with students (M. Sasakura et al., 2021).

(a) Images taken by video camera may be organized as virtual reality class (VR-class), where students can take part in the VR-class by the computing facility.

(b) The images can include not only a teacher with blackboards but also class mates.

(c) VR-class can include any angle from the point of video camera shots. Several blackboards in a class may be observed.

(d) Some zoom up to the blackboard is implemented, for the student to observe the contents written on.

In the constructed VR-class, a teacher can take a consultant role for analysis of causal relation. Then the teacher would present a chart to transform the analysis to database, where the transformation is realized as an abstract process from effectiveness in causal analysis into recursive Backus Naur Form (for structural representation of objects), as below.

At Virtual Reality Class:

Teacher — Consultant

Consultation:

$$\begin{array}{rcl} \text{Causal Analysis} & \Rightarrow & \text{Database} \\ \text{Abstract Effectiveness} & \Rightarrow & \text{Recursive BNF} \end{array}$$

Effectiveness is historically formulated as in logical frameworks. Even trends are given in studies on reductions of decidability (Rasga et al., 2021) and on what an inference is (Tennant, 2021).

In this paper, effectiveness can be captured as follows, for causal analysis:

Effects	Requisites	Prohibitions
$Flying$	$Airline$	$Infected$
...	...	...
...	...	...

For example, $Airplane$ (a predicate) and Not $Infected$ (a negated predicate) may cause the $Flying$ effect (a predicate), and so on.

As a design to realize effectiveness obtained by analysis, database is a static framework, whose dynamic behaviors are implemented as retrieval functions. Such database may be represented by recursive Backus-Naur Form (BNF), in this paper, where BNF is of use for structural analysis of recursion. For example, a list *List* over a set of objects *Ob* can be expressed in the manner like

$$L ::= null \mid cons(a, L)$$

where:

(a) *null* denotes the empty list.

(b) a is from *Ob*, and

(c) *cons* is a specific function (to take an object followed by a list and form an extended list).

In the following sections, recursive BNF and model theory for database would be made in details.

As a background, such a transformation of effectiveness (in table) to recursive BNF (as database descriptions) comes from an algebraic structure of a bounded lattice, as follows:

Effectiveness $\Longrightarrow$ Recursive BNF (Based on a Bounded Lattice)

A bounded lattice $(As, \bigvee, \bigwedge, \bot, \top)$ equipped with the partial order $\sqsubseteq$ and an implication $\Rightarrow$ may be taken:
(i) $\bot$ and $\top$ are the least and the greatest elements of the algebra (set) *As*, respectively, with respect to the partial order $\sqsubseteq$.
(ii) The least upper bound (*join*) $\bigvee$ and the greatest lower bound (*meet*) $\bigwedge$ exist for any two elements of *As*.
(iii) The implication $\Rightarrow$ (a relation on *As*) is defined in a way that $z \sqsubseteq (x \Rightarrow y)$ iff $x \bigwedge z \sqsubseteq y$.

3 MODAL LOGIC TO REPRESENT CONSULTATION

A motive of this section is to represent the meaning of consultation made by a consultant as in VR-class. The consultation is here conditioned by logical formulas such that the sense or meaning of consultation may be represented by semantics of logical formulas.

To see some adequate logical framework, consultation is now captured as abstract transformation from analysis to design, which is conceived in transitions of abstract state machine or transition system (for model theory) of modal logic.

With respect to representation of analysis (name), prefix modality is adopted. As regards representation of design (name), we take postfix modality. In addition to postfix modality, fixed point operator is needed, for consultation to denote a state set in transition system.

A modal logic with greatest fixed point operator and with postfix modality is presented, where it may be modified from modal mu-calculus (Kozen, 1983) and Henessy-Milner Logic.

The set Ψ of (logical) formulas are defined inductively as follows.

$$\psi ::= \top \mid p \mid \neg\psi \mid \psi \wedge \psi \mid \nu X.\psi \mid [a]\psi \mid \psi[d]$$

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

(a) $\top$ is the truth.

(b) p denotes propositions.

(c) $\wedge$ stands for the conjunction.

(d) $\neg$ is the logical negation.

(e) ν is a greatest fixed point operator.

(f) $[a]$ is a prefix modality with analysis label (name) a.

(g) $[d]$ is a postfix modality with design label (name) d.

(h) ν is a greatest fixed point operator, with X a proposition variable.

A Transition System $\mathcal{S}$:

For the set Ψ of formulas, a transition system $\mathcal{S}$ is defined to be (S, A, D, Re, Rel, V) where:

(i) S is a set of states.

(ii) A is a set of labels for analyses.

(iii) D is a set of labels for designs.

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

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

(vi) $Val : Prop \to 2^S$ maps to each proposition (variable) a set of states.

Semantic Functions:

A semantic function $[\![\ ]\!] : \Psi \to 2^S$ is defined to regard the semantics of a formula ψ as a subset of S by $[\![\psi]\!]$.

(a) $[\![\top]\!] = S$.

(b) $[\![p]\!] = Val(p)$.

(c) $[\![\psi_1 \wedge \psi_2]\!] = [\![\psi_1]\!] \cap [\![\psi_2]\!]$.

(d) $[\![\neg\psi]\!] = S - [\![\psi]\!]$.

(e) $[\![[a]\psi]\!]$
$= \{s \in S \mid \forall s' : (s,s') \in Re(a)$ entails $s' \in [\![\psi]\!]\}$.

(f) $[\![\psi[d]]\!]$
$= \{s \in S \mid \forall s' : (s',s) \in Rel(d)$ entails $s' \in [\![\psi]\!]\}$.

(g) $[\![\nu X.\psi]\!] = \cup \{T \subseteq S \mid T \subseteq [\![\psi]\!]_{[X:=T]}\}$,
where every occurrence of X in ψ is positive, and $[\![\psi]\!]_{[X:=T]}$ denotes a set obtained from $[\![\psi]\!]$ by substituting T for any occurrence of X, and $\cup$ stands for a union (of sets).

Note for μ-calculus including a least fixed point operator μ that, with negation sign $\neg$,

$\bot = \neg\top$,
$\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$,
$\langle a\rangle\varphi = \neg[a]\neg\varphi$, and
$\mu X.\varphi = \neg\nu X.\neg\varphi[X := \neg X]$,
where $\varphi[X := \neg X]$ means substituting $\neg X$ for X in all free occurrences of X in φ.

Semantics for Consultation:

What such a VR-class consultant denotes is here discussed. From the function views, it is regarded as a transformation, abstracted from consultation for design with analysis of effects caused by requisites and prohibitions.

Without concretized data of computing environments in a state, a state set may denote the sense of consultation, conditioned by a logical formula.

(a) The state set denoted by $[a]\psi$ is the one which is regarded as states before applying analysis, causing state transition to the state set denoted by ψ.

(b) The state set denoted by $\psi[d]$ is the one which is regarded as states after applying design, causing state transition from the state set denoted by ψ,

(c) They are both to contain the state sets denoted by ψ and by a proposition variable X (occurring in ψ) as consultation denotations.

When an environmental state set of a consultant is conditioned by a formula ϕ containing positive occurrences of a proposition variable X, the transformation of analysis a to design d is an abstraction of consultation, whose semantics may be given as a state set $[\![X]\!]$ such that

$$[\![X]\!] = [\![[a]\phi(X) \wedge \phi(X)[d]]\!],$$

where $\phi(X)$ is just ϕ with positive occurrences of X.

Because the empty state set $\emptyset$ satisfies this equation, but too trivial to represent such an environmental aspect of semantics. Rather than the least fixed point, we can have a greatest fixed point of the equation, which can be expressed by ν-operator:

$$[\![\nu X.[a]\phi \wedge \phi[d]]\!]$$

We may also make use of $[\![\nu X.[a]\phi]\!] \cap [\![\nu X.\phi[d]]\!]$, because of Proposition 1.

Proposition 1. *We have:*

$$[\![\nu X.[a]\psi \wedge \psi[d]]\!] \subseteq [\![\nu X.[a]\psi]\!] \cap [\![\nu X.\psi[d]]\!].$$

Proof. With respect to the greatest fixed point of monotonic function, we see

$$[\![\nu X.[a]\psi \wedge \psi[d]]\!] \subseteq [\![\nu X.[a]\psi]\!].$$

Similarly it is seen that

$$[\![\nu X.[a]\psi \wedge \psi[d]]\!] \subseteq [\![\nu X.\psi[d]]\!].$$

Thus the proposition holds. □

4 RECURSIVELY DESCRIBED DATABASE

The recursive representation transformed from effects (which are caused by requisites and prohibitions as in Section 2) is examined, where recursive representation may induce database. Abstract representation is firstly given by Backus-Naur Form for effectiveness. Then model theory is discussed with respect to abstract representation inducing database.

4.1 Abstract Representation

A set of expressions of the form $Req;Pro \Rightarrow Ef$ is considered, where Req and Pro represent requisites and prohibitions, respectively, for an effect Ef.

It can be described in Backus-Naur Form (BNF), by recursion.

$$\begin{aligned}
&Rule ::= \emptyset \mid \{rule\} \cup Rule\\
&rule ::= Con \Rightarrow Ef\\
&Con ::= Req;Pro\\
&Req ::= null_{Req} \mid q;Req\\
&Pro ::= null_{Pro} \mid not\ q : Pro\\
&Ef ::= q \mid not\ q
\end{aligned}$$

where the semicolon are used to stand for concatenations of strings, and:

(a) $\emptyset$ is the empty set in $Rule$.

(b) $null_{Req}$ is the empty sequence in Req and $null_{Pro}$ is the empty sequence in Pro, where $null_{Con} = null_{Req}$; $null_{Pro}$ is the empty sequence in Con.

(c) q is a variable for predicates.

(d) *not* is negation sign.

Concretization of BNF:

This BNF can be concretized as causal relation by means of logical or algebraic expressions. Each *rule* can represent:

$$q_1 \wedge \ldots \wedge q_n \wedge not\ r_1 \wedge \ldots \wedge not\ r_m \rightarrow p$$

where p is q or *not* r with the symbols:

(a) q, q_i $(1 \leq i \leq n)$, r, r_j $(1 \leq j \leq m)$ as predicate or algebraic variables.

(b) $\wedge$ as the logical *and* or the algebraic *meet* operation.

(c) $\rightarrow$ as logical or algebraic implication.

Rule may denote a logical *and* or algebraic *meet* of *rule*s such that *Rule* is regarded as causal relation.

3-Valued Model Theory:

Base on 2-valued model domain, answer set programming (Gebser and Schaub, 2016) and its application to problem solving (Kaufmann et al., 2016) are established. With the unknown (denoted by "1/2") as in 3-valued domain $\{0, 1/2, 1\}$, causal relation should be made clearer, for data technologies, in abstraction from logical or algebraic representations.

The domain $\{0, 1/2, 1\}$, with a partial order $\sqsubseteq$:

$$\begin{array}{l} 0 \sqsubseteq 0, 0 \sqsubseteq 1/2, 0 \sqsubseteq 1, \\ 1/2 \sqsubseteq 1/2, 1/2 \sqsubseteq 1, \\ 1 \sqsubseteq 1, \end{array}$$

with 0 as the bottom and with 1 as the top, is a bounded lattice with an implication, as introduced in Section 2. We here take this 3-valued domain with model theory in abstracted BNF.

For a given *Rule*, let

$$\Sigma_{Rule} = \{q \mid q \in Rule\}.$$

It is denoted just by Σ, if the set *Rule* is clear in the context.

With an assignment

$$V : \Sigma \rightarrow \{0, 1/2, 1\},$$

the evaluation of a set *Rule* is given in conditional statement form. The conditional statement form is

$$[v_1 \rightarrow e_1, \ldots, v_n \rightarrow e_n, e_{n+1}]\ (n \geq 1),$$

which is to describe:

$$\begin{array}{l} \text{if } v_1 \text{ then } e_1 \\ \text{else if } v_2 \text{ then } e_2 \\ \ldots \\ \ldots \\ \text{else if } v_n \text{ then } e_n \\ \text{else } e_{n+1} \end{array}$$

An evaluation function *eval* with respect to an assignment V is recursively defined for a rule set *Rule*. The evaluation is similar to the one (Yamasaki and Sasakura, 2021a), where the cases for evaluations of expressions are regarded as represented with the conditional statement form.

$$\begin{array}{l} eval_V(Rule) = \\ [\forall rule \in Rule : eval_V(rule) = 1 \rightarrow 1, \\ \exists rule \in Rule : eval_V(rule) = 0 \rightarrow 0, 1/2] \end{array}$$

$$\begin{array}{l} eval_V(rule) = \\ [eval_V(Con) \sqsubseteq eval_V(Ef) \rightarrow 1, \\ eval_V(Ef) = 0 \rightarrow 0, 1/2] \end{array}$$

$$\begin{array}{l} eval_V(Con) = \\ [\forall q \in Re : V(q) = 1 \text{ and} \\ \forall not\ q \in Pro : V(q) = 0 \rightarrow 1, \\ \exists q \in Re : V(q) = 0 \text{ or} \\ \exists not\ q \in Pro : V(q) \neq 0 \rightarrow 0, 1/2] \end{array}$$

$$\begin{array}{l} eval_V(Ef) = \\ [Ef = q \rightarrow V(q), Ef = not\ q \text{ with } V(q) = 0 \rightarrow 1, 0] \end{array}$$

$$eval_V(not\ q) = [V(q) = 0 \rightarrow 1, 0]$$

4.2 Model Theory

We here present model theory for recursively defined *Rule* in BNF where there is no case that both forms $Con \Rightarrow q$ and $Con' \Rightarrow not\ q$ exist, for some q.

Model Theory:

For a set *Rule* with an assignment V, let

$$\begin{array}{l} Pos_V = \{q \mid V(q) = 1\}, \text{ and} \\ Neg_V = \{q \mid V(q) = 0\}. \end{array}$$

Note that $Pos_V \cap Neg_V = \emptyset$, that is, the pair (Pos_V, Neg_V) is consistent. If $eval_V(Rule) = 1$, then the pair form (Pos_V, Neg_V) is interpreted as a model of *Rule*.

For a *rule* of the form $Con \Rightarrow Ef$, $Con(rule)$ stands for *Con* and $Ef(rule)$ for Ef, respectively.

Definition 2. A mapping

$$Trans : (2^\Sigma \times 2^\Sigma) \rightarrow (2^\Sigma \times 2^\Sigma)$$

is defined, such that, with *Trans* to be applied to (Pos_V, Neg_V), $(Pos_{V'}, Neg_{V'})$ may be obtained:

(a) If there exists a *rule* such that $Con(rule) = null_{Con}$ or $eval_V(Con(rule)) = 1$, $Ef(rule)$ is in $Pos_{V'}$.

(b) If there is no *rule* with $Ef(rule) = q$ and no *rule* with $Ef(rule) = not\ q$, q is in $Neg_{V'}$.

(c) For any *rule* with $Ef(rule) = q$ such that $eval_V(Con(rule)) = 0$, q is in $Neg_{V'}$.

(d) If there exists a *rule* with $Ef(rule) = not\ q$ such that $eval_V(Con(rule)) \neq 0$, q is in $Neg_{V'}$.

We describe properties of the mapping *Trans*.

Proposition 3. *Assume the mapping*

$$Trans : (2^\Sigma \times 2^\Sigma) \rightarrow (2^\Sigma \times 2^\Sigma).$$

Consistency is preserved under Trans applied to (Pos_V, Neg_V) *for an assignment* V.

Proof. For a consistent pair (Pos_V, Neg_V), let

$$(Pos_{V'}, Neg_{V'})$$

be $Trans(Pos_V, Neg_V)$, and examine any $q \in \Sigma$. With Definition 2, the case for q to be included in $Pos_{V'}$ (case (a)) and the cases for q to be included in $Neg_{V'}$ (cases (b), (c) and (d)) are exclusive. Thus V' is well defined such that $(Pos_{V'}, Neg_{V'})$ is consistent. □

Fixed Point Semantics:

A partial order $\subseteq_{com}$ on $2^\Sigma \times 2^\Sigma$ is defined:

$$(Pos_V, Neg_V) \subseteq_{com} (Pos_{V'}, Neg_{V'})$$
$$\text{iff } Pos_V \subseteq Pos_{V'} \text{ and } Neg_V \subseteq Neg_{V'}$$

The mapping $Trans$ is not monotonic in the sense that even if $(Pos_{V_1}, Neg_{V_1}) \subseteq_{com} (Pos_{V_2}, Neg_{V_2})$, it does not mean that

$$Trans(Pos_{V_1}, Neg_{V_1}) \subseteq_{com} Trans(Pos_{V_2}, Neg_{V_2}).$$

Proposition 4. *Assume a fixed point of Trans for an assignment V to the set of rules Rule,*

$$(Pos_V, Neg_V) = Trans(Pos_V, Neg_V).$$

Then $eval_V(Rule) = 1$.

Proof. For each $q \in \Sigma_{Rule}$, we check the evaluation of the set $Rule$.

(a) If $q \in Pos_V$, then

$$\forall rule : [Ef(rule) = q \rightarrow eval_V(rule) = 1].$$

(b) Let $q \in Neg_V$. If there is no $rule$ with $Ef(rule) = q$ and no $rule$ with $Ef(rule) = not\ q$, do not care this case for the evaluation of $Rule$.

(c) With $q \in Neg_V$, and for any $rule$: $[eval_V(Con(rule)) = 0$ and $eval_V(Ef(rule)) = 0]$, $eval_V(rule) = 1$.

(d) With $q \in Neg_V$, we see that for any $rule$: $Ef(rule) = not\ q$, $eval_V(not\ q) = 1$. Thus $eval_V(rule) = 1$.

(e) Assume $q \notin Pos_V \cup Neg_V$. For any $rule$: $Ef(rule) = q$, $eval_V(Con(rule)) = 0$ or $1/2$ and $eval_V(Ef(rule)) = 1/2$. It follows that $eval_V(rule) = 1$. For and $rule : Ef(rule) = not\ q$, $eval_V(Con(rule)) = 0$ and $eval_V(Ef(rule)) = 0$. Thus $eval_V(rule) = 1$.

□

Logical Database:

By Proposition 4 for $Rule$, a fixed point of $Trans$, $(Pos_{V_{fix}}, Neg_{V_{fix}})$, can be a model of $Rule$, with an assignment V_{fix}. The set $Pos_{V_{fix}}$ contains predicates to be retrieved, where the set $Neg_{V_{fix}}$ contains predicates to be negated by being not retrieved. Not all retrieval predicates may be obtained from any $Rule$, however, if a fixed point exists, retrievals are available in logical database represented by BNF, even with strong negation. Thus the BNF can be regarded as a design to represent analysis of effectiveness.

The model pair

$$(Pos_{V_{fix}}, Neg_{V_{fix}})$$

corresponds to the retrieval scheme for the set $Rule$ as database:

In the sense that

$q \in Pos_{V_{fix}} \Rightarrow q$ is retrieved,
$q \in Neg_{V_{fix}} \Rightarrow q$ is not retrieved, and
$q \notin Pos_{V_{fix}} \cup Neg_{V_{fix}} \Rightarrow$ retrieval of q is undefined,

the scheme is implemented. That is, for each query q to database expressed by BNF, retrieval result is expected to answer

(i) Yes (if $q \in Pos_{V_{fix}}$),

(ii) No (if $q \in Neg_{V_{fix}}$) and

(iii) Unknown (otherwise).

Query q (predicate) $\longrightarrow$	Recursive BNF *Rule*
Yes/No/Unknown $\longleftarrow$	(as Database)

4.3 Negation as Failure

To relax complexity caused by nonmonotonic mapping $Trans$, we focus on strong negation of the form *not q*, to take the approximation that default or negation as failure may be substituted for strong negation.

The negation as failure is classically established: If some predicate p cannot be derived from the set of formulas Γ in a proof system, then p may be negated:

$$\Gamma \nvdash p \Rightarrow \neg p.$$

Approximate Evaluation on the Restricted Rule:

$approx_V$ is defined recursively, with respect to an assignment V for $Rule$, where each $rule$ of $Rule$ contains only the positive of the form q as $Ef(rule)$, that is, each rule $rule$ does not contain negation sign in $Ef(rule)$.

$approx_V(Rule) =$
$[\forall rule \in Rule : approx_V(rule) = 1 \rightarrow 1,$
$\exists rule \in Rule : approx_V(rule) = 0 \rightarrow 0, 1/2]$

$approx_V(rule) =$
$[approx_V(Con) \sqsubseteq approx_V(Ef) \rightarrow 1,$
$approx_V(Ef) = 0 \rightarrow 0, 1/2]$

$approx_V(Con) =$
$[\forall q \in Re : V(q) = 1$ and
$\forall not\ q \in Pro : V(q) = 0 \rightarrow 1,$
$\exists q \in Pro : V(q) = 0$ or
$\exists not\ q \in Pro : V(q) = 1 \rightarrow 0, 1/2]$

$approx_V(Ef) = V(q)$ for $Ef = q$

$approx_V(not\ q) =$
$[V(q) = 0 \rightarrow 1, V(q) = 1 \rightarrow 0, 1/2]$

We thus substitute negation as failure for strong negation, in evaluation of *Rule*.

An evaluation function $approx_V$ is then analyzed in the following.

Definition 5. A mapping

$$Tr : 2^{\Sigma} \times 2^{\Sigma} \rightarrow 2^{\Sigma} \times 2^{\Sigma}$$

is defined, such that, with Tr to be applied to $(suc_V, fail_V)$, $(suc_{V'}, fail_{V'})$ may be obtained:

(a) If there exists a *rule* such that $Con(rule) = null_{Con}$ or $approx_V(Con(rule)) = 1$, $Ef(rule)$ is in $suc_{V'}$.
(b) For any $rule : approx_V(Con(rule)) = 0$, $Ef(rule)$ is in $fail_{V'}$.

Proposition 6. *Assume the mapping*

$$Tr : (2^{\Sigma} \times 2^{\Sigma}) \rightarrow (2^{\Sigma} \times 2^{\Sigma}).$$

Consistency is preserved under Tr applied to $(suc_V, fail_V)$ *for an assignment* V.

Proof. Let $(suc_{V'}, fail_{V'}) = Tr(suc_V, fail_V)$. If $(suc_V, fail_V)$ is consistent, the cases for $Ef(rule)$ to be included in $suc_{V'}$ and to be included in $fail_{V'}$ are mutually exclusive. Therefore the pair $(suc_{V'}, fail_{V'})$ is consistent, so that consistency may be preserved under the mapping Tr. □

As regards comparison of the evaluation $eval_V$ with $approx_V$, we have:

Proposition 7. *For any "rule" of "Rule",*

$$eval_V(Con(rule)) \sqsubseteq approx_V(Con(rule)),$$

and $eval_V(Ef(rule)) = approx_V(Ef(rule))$.

Proof. It is because $eval_V(not\ q) \sqsubseteq approx_V(not\ q)$, for $Con(rule)$, where
$eval_V(Ef(rule)) = approx_V(Ef(rule))$
for $Ef(rule)$ containing no negation sign. □

For simplicity of treatment of the mapping Tr, let $TR : \Omega \rightarrow \Omega$ be defined for the set

$$\Omega = \{V \mid V : \Sigma \rightarrow \{0, 1/2, 1\} \text{ is an assignment } \}$$

with respect to $\Sigma = \Sigma_{Rule}$ such that

$$TR(V_1) = V_2 \text{ iff}$$
$$Tr(suc_{V_1}, fail_{V_1}) = (suc_{V_2}, fail_{V_2}).$$

Definition 8. A partial order $\sqsubseteq_{assign}$ on the set Ω of assignments is defined: $V \sqsubseteq_{assign} V'$ if, for two pairs $(suc_V, fail_V)$ and $(suc_{V'}, fail_{V'})$,

$$(suc_V, fail_V) \subseteq_{com} (suc_{V'}, fail_{V'}).$$

The mapping TR is monotonic in the following sense.

Proposition 9. *If* $V_1 \sqsubseteq_{assign} V_2$, *then*

$$TR(V_1) \sqsubseteq_{assign} TR(V_2).$$

Proof. With the correspondence of the mapping TR to Tr, we need that if $(suc_{V_1}, fail_{V_1}) \subseteq_{com} (suc_{V_2}, fail_{V_2})$, then

$$Tr(suc_{V_1}, fail_{V_1}) \subseteq_{com} Tr(suc_{V_2}, fail_{V_2}).$$

By the definition of $approx_V$, we can see this. □

Because the mapping TR is monotonic, there is a (least) fixed point of TR.

Proposition 10. *There is a fixed point* V_0 *of* TR *such that* $TR(V_0) = V_0$.

Proof. Following the ordinary monotonic function theory on the complete lattice, we see that

$$\sqcap\{V \in \Omega \mid TR(V) \sqsubseteq_{assign} V\}$$

is a (least) fixed point of TR, where $\sqcap$ stands for the greatest lower bound. □

With such a fixed point assignment V_0, we have:

Proposition 11. *For the restricted Rule and with a fixed point* V_0 *of* TR,

$$approx_{V_0}(Rule) = 1 \text{ and } eval_{V_0}(Rule) = 1.$$

Proof. It is known that $V_0 = TR^{\alpha}(\emptyset_{\Omega})$ for some ordinal α, where $TR^{\beta}(\emptyset_{\Omega}) = TR(TR^{\beta-1}(\emptyset_{\Omega}))$ for β: successor ordinal, or $\sqcup_{\gamma<\beta} TR^{\gamma}(\emptyset_{\Omega})$ for β: limit order (with the least upper bound notation $\sqcup$), where:
(i) $\emptyset_{\Omega} \in \Omega$ denotes $(\emptyset, \emptyset) \in 2^{\Sigma} \times 2^{\Sigma}$.
(ii) TR preserves consistency, and TR is inclusive such that TR^{α} preserves consistency.

It is seen that $approx_{\emptyset_{\Omega}}(rule) = 1$. In addition, for any *rule*,

$$approx_{\emptyset_{\Omega}}(rule) \sqsubseteq approx_{TR^{\beta}(\emptyset_{\Omega})}(rule),$$
$$\text{with assignments } \emptyset_{\Omega} \sqsubseteq_{assign} TR^{\beta}(\emptyset_{\Omega}).$$

Thus $approx_{V_0}(rule) = 1$ for any *rule* such that $approx_{V_0}(Rule) = 1$. With Proposition 7, for any *rule*,

$$eval_{V_0}(Con(rule)) \sqsubseteq approx_{V_0}(Ef(rule)) = eval_{V_0}(Ef(rule)).$$

It follows that $eval_{V_0}(rule) = 1$ for any $rule$, and thus $eval_{V_0}(Rule) = 1$. □

5 CONCLUSION

This paper deals with abstraction of a teacher's analysis and synthesis such that the role of a consultant may be described. If causal relation is analyzed by a consultant to design a system, the abstract consultation may be understood as a transformation of effectiveness caused by analysis to design of representation for database.

The primary result of this paper is to formulate the role of a consultant as abstract consultation which can be described in modal logic with the greatest fixed point operator, even on the propositional base.
(i) The modal logic of this paper involves prefix modality for analysis and postfix modality for design, where both analysis and design are called by names.
(ii) Conditioning the behaviors to receive analysis by prefix modality and to provide design by postfix modality, we have formulas in our modal logic. The denotation of a formula is presented by a state set, where states virtually keep computing environments, and state transitions are meaningful as relations regarding modal operators.
(iii) With respect to an equilibrium, a state set is given as a greatest fixed point of the denotation for a formula conditioned to the behavior of consultation.

As the secondary result, we have a model theory in 3-valued domain for the representations by Backus-Naur Form as designed database (corresponding to given effects as causal relations).
(i) Different from those in problem solving by answer set programming, model theory in 3-valued logic conceives some hard problem. We defined a fixed point semantics by some mapping associated with a given BNF representation. It is a basis of retrieval in the database to be represented by Backus-Naur Form.
(ii) It is not always the case that we could see a model of any representation of database. In such model theory, an approximate idea of strong negation by negation as failure is provided. In the case of negation as failure (default) instead of strong negation, model theory is easier for some restricted class of representations as database. The case with communication facilities (Yamasaki and Sasakura, 2021b) is theoretically relevant to the present case.

REFERENCES

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

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

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

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

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

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

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

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.

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

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

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

Naumov, P. and Tao, J. (2019). Everyone knows that some knows: Quantifiers over epistemic agents. *Review.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.

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

Yamasaki, S. and Sasakura, M. (2020). Modal mu-calculus extension with description of autonomy and its algebraic structure. In *Proceedings of the 5th International Conference on Complexity, Future Information Systems and Risk*, pages pages 63–71.

Yamasaki, S. and Sasakura, M. (2021a). 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. (2021b). Distributed strategies and managements based on state constraint logic with predicate for communication. In *Proceedings of the 6th International Conference on Complexity, Future Information Systems and Risk*, pages 78–85.