# Distributed Strategies and Managements based on State Constraint Logic with Predicate for Communication

Susumu Yamasaki and Mariko Sasakura
*Department of Computer Science, Okayama University, Tsushima-Naka, Okayama, Japan*

Keywords: Logical Approach, Organization of Strategy, Logic for Distributed Systems, Model Theory.

Abstract: From the views on cognitive management, this paper deals with state constraint and distributed systems, where communication between states is a key function of complexity. The primary purpose is concerned with logical analysis of complex, distributed system structure which contains strategies (procedures) designed in states. Between states, strategies may be communicative and transferrable, where the transferrability is supposed to be given by predicates for communication between states. The strategy as a procedure is assumed to be inductively constructed by other distributed strategies. The structure to represent the designed way of strategies takes an inductively defined form, on which some logical relation is characterized with respect to the compound construction of strategies (procedures). The logical relation is in accordance with possibly infinite set of propositional formulas constrained by states. As regards procedural executions, implementation, the undefined (implementation), and non-implementation may be considered for the remarked strategy. Based on the discussions of implementability for strategic constructions, a structural analysis of distributed strategies may be settled as 3-valued model theory of logical expressions. It is related to 3-valued model theory, where some fixed point theory is now examined, with respect to the mapping (which is in general monotonic) associated with a logical expression. The logical expression of this paper can be denoted as a propositional logic formula with default negation. As an application to logical system, logical formulas with both strong and default negations may be analyzed with 3-valued domain. This paper thus abstracts application of logical expressions to structural analysis of distributed strategies. Structure of strategies is complex owing to distribution of state constraint strategies, but effectiveness may be endowed with logical approach and abstraction of communication facility.

## 1 INTRODUCTION

Cognitive management subjects include complexity analysis of distributed systems containing computing and communication facilities. As such a distributed system, we consider an *abstract state machine* frmawork, where the states are distributed and strategies (as procedures) are integrated into each state. Between states, communications and behavioral interactions are allowed, as well as state transitions which are caused by strategic actions in states. As strategies in abstract state machine possibly for data operations, there are procedural, logical and algebraic views.
(1) Procedural strategy is expressed by denotational approach in the book (Mosses, 1992). The procedural method is in accordance with operational implementation for programs to be executed. The strategies may be abstracted, with functional programs (Bertolissi et al., 2006). These are essentially programs on data, being interpreted as data operations.
(2) Strategies are also captured in logical systems from the viewpoints of sequential process, as in the papers (Giordano et al., 2000; Hanks and McDermott, 1987). These are concerned with dynamism of actions, and modality in logical systems. The action is formulated as a key role in strategic reasoning of abstract state machine, as well as concretized actions in dynamic logic. Acting and sensing failures are discussed as advanced works (Spalazzi and Traverso, 2000).
(3) Primarily from the viewpoints of transitions, abstract state machine is discussed, in the paper (Reps et al., 2005). We can have specified structure of streams caused by abstract state transitions, by the note (Rutten, 2001). Algebraic structures are formally discussed with respect to state transitions (Droste et al., 2009).

With relevance to practical senses of strategies, AI reasoning may be common interests even in distributed system designs. With the concerns to



Yamasaki, S. and Sasakura, M.
Distributed Strategies and Managements based on State Constraint Logic with Predicate for Communication.
DOI: 10.5220/0010468700780085
In *Proceedings of the 6th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2021)*, pages 78-85
ISBN: 978-989-758-505-0

AI programming, we have studied nonmonotonic reasoning where fixed point theory is not alway a routine. In knowledge representations with respect to logical approaches to reasoning, we have seen backgrounds:
(1) Logics with knowledge (Reiter, 2001) are classical, where knowledge (data) representation and reasoning as operation in knowledge base are systemized in logics.
(2) "Distributed knowledge" is discussed (Naumov and Tao, 2019), with quantified variables of quantifies ranging over the set of agents. Concerning applications of the second-order predicates to knowledge, the paper (Kooi, 2016) contains the concept of knowing.
(3) Regarding distributed systems, software and knowledge engineerings like mobile ambients (Cardelli and Gordon, 2000; Merro and Nardelli, 2005) have been formulated with environments to make communication reasonable. As proofs in programming and data science, the papers (Dam and Gurov, 2002; Kozen, 1983) are classical enough to formulate the proof systems with fixed points and their approximations.

Following the ideas of strategies in abstract state machine, and AI reasonings, this paper aims at structural aspects of strategies as procedures which may be basis possibly for AI programming with algebra or logic, rather than the whole algebraic structure of abstract state machine. This paper deals with a framework, different from those established works, and examines distributed strategies just with respect to inductive structure of strategy construction, where strategies are inductively defined in various states, and integrated into each state. Between states, strategies are communicative and transferrable, where each state contains strategies on data. For descriptions of strategy constructions in a distributed system, possibly infinite set of propositional logic formulas is adopted in the sense that it may express abstract and clear structures. With universal denotations for possibly infinite set of propositional logic formulas (which may represent first-order logic formula with Herbramd base), we have effectiveness (with respect to computing) for the system construction of this paper. As regards implementability of strategies with communication between states in a distributed system, 3-valued model theory of logical expressions provides analysis with default negation (in accordance with negation as failure) and the unknown for implementability. Abstracted from the structural implementability of strategy constructions, logical expressions constrained by states (virtually with intensionality sensitive to states and communication between states) may be formulated for a distributed system.

As a primary goal of this paper, 3-valued model theory of logical expressions is given. The predicates of implementability and default negative are next organized. As an application, we examine logical database with query predicates where we could have model theory of propositional logic with both strong and default negations, and with communication facility. The paper is organized as follows. Section 2 presents an outlook on inductive structure of strategy constructions from the view of distributed resources. In Section 3, logical expressions and model theory are discussed, abstracted from implementability of strategy constructions. Section 4 presents the predicates of evaluations for expressions and derivations as query of strategy (procedure) implementability, with reference to application to logical database, where strategy names are regarded as organizing database. Concluding remarks are described in Section 5.

# 2 ORGANIZATION OF STRATEGIES

## 2.1 State Constraint Objects

With respect to process theories (Hennessy and Milner, 1985; Kucera and Esparza, 2003; Milner, 1999), we consider the case that state constraint objects are distributed but relational, where the case conceives complexity in managements for cognitive or computing merits, but representations by means of distributed objects are easily available as long as communication between objects is implicitly guaranteed. This paper treats such objects distributed and constrained by states with manageable relations.

As an illustration, we present state constraint propositions represented as adjective for strategic senses:

(i) $secure[h]$, $remedied[h]$, $safe[h]$ and $risky[h]$ at a state $h$ (as regards health-affair), and
(ii) $bound[s]$ and $open[s]$ at a state $s$ (as regards social-affair),
where logical *and* "$\wedge$" and *implication* "$\rightarrow$" are used as well as *negation* "$\sim$", in addition to parentheses for discriminations.

$h$-state:
$secure[h]$
$\wedge(\sim remedied[h] \wedge \sim safe[h] \rightarrow risky[h])$
$\wedge(\sim open[s] \rightarrow \sim safe[h])$

$s$-state:
$(secure[h] \rightarrow bound[s])$
$\wedge(bound[s] \wedge \sim risky[h] \rightarrow open[s])$

Communication of $s$-state might be supposedly available to $h$-state such that the entailed propositions can be acknowledged at state $h$. When the proposition $secure[h]$ is assumed, then what reasoning may be taken into consideration, on the basis of propositional logic constrained by states. The negation can be thought of as default (i.e., negative as failure of reasoning, in 3-valued logic containing the unknown to be reasoned or not to be reasoned.)

Assume no communication of $h$-state to $s$-state (i.e. neglect of $h$-state propositions at $s$-state) except $secure[h]$: (i) At state $s$, with $secure[h]$, $bound[s]$ is inferred. Even with $bound[s]$, ("without" default $\sim risky[h]$,) $open[s]$ is unknown. (ii) At state $h$, $\sim safe[h]$ is unknown from unknown $\sim open[s]$, such that even with default $\sim remedied[h]$, $risky[h]$ may be unknown, and even the default $\sim risky[h]$ may not transferrable to state $s$.

Without implementation details of communication, we may see reasoning from declarations of logical formulas of state constraint propositions, which is regarded as coordination of computing (based on reasoning) with communication. In this paper, we deal with such a coordination for distributed (propositional) variables (at states), which can represent procedural or strategic objects by name.

## 2.2 Representation of Strategy Constructions

As a formal system for distributed programming with state environments, we are interested in strategic programming where strategies as procedures are distributed such that each strategy may be compiled into construction with other distributed strategies. Then the structure of strategy constructions should be represented in a simpler form than verbal accounts with refined procedural words. In case of problem solving, the strategy is called by name and the reference structure is based on recursion manners of goals, constructed by subgoals (such that the subgoal may be constructed by subgoals). We thus make the examinations of the expression for the ways of (i) how to represent the inductive structure of distributed strategy constructions, and (ii) how to represent the constructive structure of strategies.

As in object-oriented programming language, a procedure can be designed in a class, such that a form

$$\{pr_1[o_1], \ldots, pr_n[o_n]\} \rhd pr[o]$$

may be taken to see that the strategy $pr$ in the class $o$ may be inductively constructed, to contain the strategies $pr_1$ (in the class $o_1$), ..., and $pr_n$ (in the class $o_n$) (as components).

This paper treats a logical approach to represent structures of a distributed system involving strategies as procedures at states:
(a) Propositional variables denoting strategies are distributed, depending on the state variables.
(b) Some standard form of distributed strategy (procedure) constructions is assumed, and the form would be logically described.
(c) Virtual communications between states are definable by predicates containing propositional variables, such that a strategy (procedure) may be regarded as transferrable from a state to another.

# 3 STATE CONSTRAINT LOGIC WITH COMMUNICATIONS

## 3.1 Formal Description of Distributed Strategy

Following Motague grammar (by R.Montague), intentionality is defined in the manner of $\lambda s.p$ for a state variable $s$ within the scope of $\lambda$-notation to the proposition $p$. We here make use of the extension in the manner of $p[s]$, rather than intension. We then make a description of formal system in terms of logic in coordination to communication between states. Now let us see a sequence of the procedural constructions including state constraints (like class-constraints as in subsection 2.2):

$$\{pr_1[s_1], \ldots, pr_n[s_n]\} \rhd pr[s],$$
$$\ldots,$$
$$\ldots,$$
$$\{pr'_1[s'_1], \ldots, pr'_m[s'_m]\} \rhd pr'[s'],$$

with strategy (procedure) names $pr_1[s_1]$, ... , $pr[s]$, $pr'_1[s'_1]$, ... , $pr'[s']$.

The left hand of the composing $\rhd$ is referred to by *body* and the right hand is referred to by *head*. The *body* consists of (none or) finitely many expressions of the form $pr[s]$ (with or without negation), where $pr$ is a procedure name with a state $s$. The *head* consists of an expression of the form $pr[s]$: The procedure may be represented by proposition or its negative, constrained by a state. Because the procedural implementation is sensitive to logical values, if calling by name for procedures is adopted where procedural structures may be of sense. Effectiveness (for computing) of such structures may be guaranteed by in-

finiteness of propositional logic, denoting first-order logic based on Herbrand base.

With communication applied to logic as coordination of logic to communication, this paper treats the abstract strategy constrained by the set $S$ of states as well as communicative predicates. From semantics views, strategic structure might be logically defined, while from formal description viewpoints, they are defined in Backus Naur Form (BNF) as follows:

Therefore, to describe the structure "$body \rhd head$" (just with a symbol $\rhd$) referred to as a rule, we take BNF of:

(a) $literal ::= p[s] \mid \sim p[s]$
(b) $head ::= literal$
(c) $body ::= \{\,\} \mid \{literal\} \cup body$
(d) $rule ::= body \rhd head$

where (i) the notation "$\sim$" is reserved for the negative sign, (ii) $p$ is a propositional variable and $s$ is a state variable, and (iii) $\cup$ denotes the set union operation, applicable to the empty $body$, $\{\,\}$.

The rule $body \rhd head$ is contained by the whole strategy, which is a sequence of such structures as rules. Instead of the sequence, the whole structure (*Strategy* in BNF) may be alternatively defined as a finite or denumerably infinite set of such structures.

(d) $Strategy ::= \emptyset \mid \{rule\} \cup Strategy$

where (i) $\emptyset$ stands for the empty set, and (ii) as *Condition* for *Strategy*, at most one of any complementary pair $(p[s], \sim p[s])$ occurs at heads of rules.

Note that this *Condition* is in accordance with the sense that the head of a rule may denote a constructed strategy by name such that both positive and negative are not needed. Throughout this paper, this condition is assumed even without mentioning.

Then a set of rules (defined as *Strategy*) is considered as the whole structure, where its inductive structure is to be interpreted. For the interpretation, communications between states must be included, since strategies in *Strategy* contain state constraint propositions to represent implementations of distributed procedures.

As coordination of strategies with communications, we just make use of higher-order predicates of the form and a set *Commu*:

(e) $Commu ::= \emptyset \mid \{commu(s', s, p[s'])\} \cup Commu$

where *commu* is a predicate symbol, with state variables $s'$ and $s$ as well as state constraint propositional variable $p[s']$, such that $commu(s, s, p[s])$ is supposedly included in *Commu* for any $p[s]$.

Finally we have a set of programs to be organized, where their inductive structures of implementations are expressed in terms of call by name and universalities fo computing and communication may be guaranteed in the first-order logic.

(f) $Program ::= Strategy \cup Commu$

For the structural interpretation with respect to *Program*, we adopt 3-valued logic with default negation. With reference to implementability for the structure of procedural constructions, whether some procedure is implementable or non-implementable is to be noted. This paper deals with the implementability case of procedure to be undefined, so that 3-valued domain may be taken for implementability.

Assume in 3-valued domain (underlying set) $\{0, 1/2, 1\}$ that (i) the implication is based on (possibly infinite) propositional logic, and (ii) the evaluation of $\sim\ p[s]$ follows the way:

| $p[s]$ | $\sim p[s]$ |
|---|---|
| 1 | 0 |
| 1/2 | 1/2 |
| 0 | 1 |

*Example* 1. Take the logical expressions at states $h$ and $s$ of subsection 2.1. Then as *Strategy*, we can have a set:

$$\begin{aligned} Strategy \\ = \{\quad & \{\,\} \rhd secure[h], \\ & \{\sim remedied[h], \sim safe[h]\} \rhd risky[h], \\ & \{\sim open[s]\} \rhd \sim safe[h], \\ & \{secure[h]\} \rhd bound[s], \\ & \{bound[s], \sim risky[h]\} \rhd open[s] \quad \} \end{aligned}$$

As *Commu*, we assume:

$$\begin{aligned} & Commu = \\ & \{\ commu(s,s,bound[s]), commu(s,s,open[s]), \\ & \ \ commu(s,h,open[s]), commu(h,h,secure[h]), \\ & \ \ commu(h,h,safe[h]), commu(h,h,remedied[h]), \\ & \ \ commu(h,h,risky[h]), commu(h,s,secure[h])\ \} \end{aligned}$$

## 3.2 Evaluation of Strategy Implementation

The assignment of the values in 3-valued domain $\{0, 1/2, 1\}$ to the propositional variable with state constraint causes the program to have the value 0, 1/2. or 1 for implementability. To define such an interpretation of the program implementation, we consider a mapping associated with a given program to be represented in terms of logic with predicates for communication. The mapping is for a fixed point semantics which can be an interpretation of the program structure inductively constructed in subsection 3.1. Such semantics would be related to retrieval in

logical database expressed by the program with state constraints and with predicates for communication. The logical database will be shown in Section 4.

Note that the predicates of the form

$$commu(s', s, p[s'])$$

(in *Commu*) are supposedly included in *Program* (in subsection 3.1), with name $P$. Let $A_P$ (or $A$ when $P$ is explicitly supposed) be $\{p[s] \mid p[s] \in literal\}$, a set of state constraint variables occurring in $P$.

For $exp ::= literal \mid body \mid rule \mid Program$ and a pair $(I,J) \in 2^A \times 2^A$, to have the evaluation of the expression in 3-valued domain, we define a valuation $Val : exp \to 2^A \times 2^A \to \{0, 1/2, 1\}$ in the following manner, with respect to (possibly infinite) propositional logic with state constraint.

$$\text{(a) } Val[\![p[s]]\!](I,J) = \begin{cases} 1 & \text{if } p[s'] \in I \text{ with } commu(s', s, p[s']) \\ 0 & \text{if } p[s'] \in J \text{ with } commu(s', s, p[s']) \\ 1/2 & \text{otherwise} \end{cases}$$

$$\text{(b) } Val[\![\sim p[s]]\!](I,J) = \begin{cases} 1 & \text{if } p[s'] \in J \text{ with } commu(s', s, p[s']) \\ 0 & \text{if } p[s'] \in I \text{ with } commu(s', s, p[s']) \\ 1/2 & \text{otherwise} \end{cases}$$

$$\text{(c) } Val[\![body]\!](I,J) = \begin{cases} 1 & \text{if } Val[\![literal]\!](I,J) = 1 \text{ for any } literal \text{ of } body \\ 0 & \text{if } Val[\![literal]\!](I,J) = 0 \text{ for some } literal \text{ of } body \\ 1/2 & \text{otherwise} \end{cases}$$

(with *Commu*)

$$\text{(d) } Val[\![rule]\!](I,J) = \begin{cases} 1 & \text{if } Val[\![body]\!](I,J) \text{ is less than or equal to } Val[\![head]\!](I,J) \\ 0 & \text{otherwise} \end{cases}$$

(for $rule = body \rhd head$ with *Commu*)

$$\text{(e) } Val[\![Program]\!](I,J) = \begin{cases} 1 & \text{if } Val[\![rule]\!](I,J) = 1 \text{ for any } rule \text{ of } Strategy \\ 0 & \text{if } Val[\![rule]\!](I,J) = 0 \text{ for some } rule \text{ of } Strategy \\ 1/2 & \text{otherwise} \end{cases}$$

(for $Program = Strategy \cup Commu$)

If $Val[\![P]\!](I,J) = 1$ for a given program $P$ and a pair $(I,J) \in 2^A \times 2^A$, such that $I \cap J = \emptyset$ (i.e., the pair $(I,J)$ is consistent), then $(I,J)$ is called a model of $P$.

With the name $P$ (of *Program*), a mapping $Tr_P$ is defined, such that its fixed point may be a model of $P$, that is, an evaluation of $P$ as 1. The model is regarded as presenting consistency of $P$ which may denote strategy constructions with communication.

The mapping $Tr_P$ (associated with a program $P$), applied to a pair $(I_1,J_1)$ for providing a pair $(I_2,J_2)$, is defined in the manner as follows.

$$Tr_P : 2^A \times 2^A \to 2^A \times 2^A,$$
$$Tr_P(I_1,J_1) = (I_2,J_2).$$

*Definition of* $Tr_P$:
(1) For some rule $body \rhd p[s]$ such that (i) for any $q[s']$ of *body*, $q[s']$ is in $I_1$ with $commu(s', s, q[s'])$, and (ii) for any $\sim r[s'']$ of *body*, $r[s'']$ is in $J_1$ with $commu(s'', s, r[s''])$, $p[s]$ is in $I_2$.
(This case contains the one that *body* is the empty set, where $p[s]$ is in $I_2$.)
(2)(a) For any rule of the form $body \rhd p[s]$ such that (i) for some $q[s']$ of *body*, $q[s']$ is in $J_1$ with $commu(s', s, q[s'])$, or (ii) for some $\sim r[s'']$ of *body*, $r[s'']$ is in $I_1$ with $commu(s'', s, r[s''])$, $p[s]$ is in $J_2$.
(This case contains the one that there is no rule of the form $body \rhd p[s]$ without any rule of the form $body' \rhd \sim p[s]$, where $p[s]$ is in $J_2$.)
(b) For some rule $body \rhd \sim p[s]$ such that (i) for any $q[s']$ of *body*, $q[s']$ is in $I_1$ with $commu(s', s, q[s'])$, and (ii) for any $\sim r[s'']$ of *body*, $r[s'']$ is in $J_1$ with $commu(s'', s, r[s''])$, $p[s]$ is in $J_2$.
(This case contains the one that *body* is the empty set, where $p[s]$ is in $J_2$.)

*Fixed Point of* $Tr_P$:
If $Tr_P(I,J) = (I,J)$, then $(I,J)$ is called a fixed point of $Tr_P$. By "componentwise subset inclusion $\subseteq_c$" (a binary relation on $2^A \times 2^A$), we mean that $I_1 \subseteq I_2$ and $J_1 \subseteq J_2$ iff $(I_1,J_1) \subseteq_c (I_2,J_2)$. When $Tr_P(I,J) \subseteq_c (I,J)$, $(I,J)$ is a prefixpoint of $Tr_P$. A fixed point of $Tr_P$ is a prefixpoint. We will see that for a fixed point $(I,J)$, if $I \cap J = \emptyset$ (i.e., $(I,J)$ is consistent), then $(I,J)$ can be a model of $P$, that is, $P$ is evaluated as 1 by the pair $(I,J)$.

*Fixed Point Models*:
The mapping $Tr_P$ is monotonic, that is, if $(I_1,J_1) \subseteq_c (I_2.J_2)$, then $Tr_P(I_1,J_1) \subseteq_c Tr_P(I_2,J_2)$. The method by fixed point of $Tr_P$ is always available as a modelling of the given program $P$.

*Example* 2. Assume the *Strategy* and *Commu* as in Example 1:

$$\begin{aligned} Strategy \\ = \{ \quad & \{\,\} \rhd secure[h], \\ & \{\sim remedied[h], \sim safe[h]\} \rhd risky[h], \\ & \{\sim open[s]\} \rhd \sim safe[h], \\ & \{secure[h]\} \rhd bound[s], \\ & \{bound[s], \sim risky[h]\} \rhd open[s] \ \ \} \end{aligned}$$

$Commu =$
$\{\ commu(s,s,bound[s]), commu(s,s,open[s]),$
$commu(s,h,open[s]), commu(h,h,secure[h]),$
$commu(h,h,safe[h]), commu(h,h,remedied[h]),$
$commu(h,h,risky[h]), commu(h,s,secure[h])\ \}$

Then a pair $(\{secure[h], bound[s]\}, \{remedied[h]\})$ may be a fixed point of $Tr_P$ ($P = Strategy \cup Commu$).

**Proposition 1.** *Assume a pair $(I,J) \in 2^A \times 2^A$ for a given program P. If a pair $(I,J)$ is a consistent fixed point of the mapping $Tr_P$, then $(I,J)$ is a model of P.*

*Proof.* Let $Tr_P(I,J) = (I_0,J_0)$. Following the definition of the mapping $Tr_P$, we examine the mapping case by case, to see why $(I_0,J_0) = (I,J)$ causes $(I,J)$ to be a model.
(1) If $p[s] \in I_0 = I$ for some rule $body \rhd p[s]$, then $Val[\![p[s]]\!](I,J) = 1$ such that
$Val[\![body' \rhd p[s]]\!](I,J) = 1$
for any rule $body' \rhd p[s]$.
(2) (i) If $p[s] \in J_0 = J$, then there may the case: for any rule of the form $body \rhd p[s]$, $q[s']$ is in $J$ with $commu(s',s,q[s'])$ for some $q[s']$ of $body$, or $r[s'']$ is in $I$ with $commu(s'',s,r[s''])$ for some $\sim r[s'']$ of $body$ (i.e., $Val[\![body]\!](I,J) = 0$) such that
$Val[\![body \rhd p[s]]\!](I,J) = 1$
for any rule $body \rhd p[s]$. (This case contains the one that there is no rule of the form $body \rhd p[s]$ for $p[s]$ without any rule of the form $body' \rhd \sim p[s]$.)
(ii) If $p[s] \in J_0 = J$, then there may be the case: for some rule of the form $body \rhd \sim p[s]$, $q[s']$ is in $I$ with $commu(s',s,q[s'])$ for any $q[s']$ of $body$, and $r[s'']$ is in $J$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$ (i.e., $Val[\![body]\!](I,J) = 1$) such that
$Val[\![body' \rhd \sim p[s]]\!](I,J) = 1$
for any rule $body' \rhd \sim p[s]$.
(3) If $p[s] \notin I_0 \cup J_0 = I \cup J$, then for any rule $body \rhd p[s]$ or any rule $body \rhd \sim p[s]$, $Val[\![body]\!](I,J) = 0$, or $1/2$ with the pair $(I,J)$ (and $Commu$). Since $Val[\![p[s]]\!](I,J) = 1/2$ and $Val[\![\sim p[s]]\!](I,J) = 1/2$,
$Val[\![body \rhd p[s]]\!](I,J) = 1$.
Thus all the rules are evaluated as 1, with respect to the pair (a fixed point of $Tr_P$) $(I,J)$. This may conclude the proposition. □

# 4 ANALYSIS OF LOGIC WITH COMMUNICATION

## 4.1 Predicates for Implementability of Strategy

To make the sense of the mapping $Tr_P$ less complex from implementation views, we have simple predicates to relate the mapping $Tr_P$ with. The predicates (of higher-order for propositions) $imple_P(p[s])$ and $default_P(p[s])$ are definable. Formally, the predicates are defined inductively, for a given program $P$ with the communicative predicates $Commu$ (shown in the subsection 3.1):

*Predicates $imple_P$(-) and $default_P$(-):*
(1) If there is a rule $body \rhd p[s]$ such that $imple_P(q[s'])$ with $commu(s',s,p[s'])$ for any $q[s']$ of $body$, and $default_P(r[s''])$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$, then $imple_P(p[s])$.
(This case contains the one that $body$ is the empty set, where $imple_P(p[s])$.)
(2)(a) If for any rule $body \rhd p[s]$ such that $default_P(q[s'])$ with $commu(s',s,q[s'])$ for some $q[s']$ of $body$, or $imple_P(r[s''])$ with $commu(s'',s,r[s''])$ for some $\sim r[s'']$ of $body$, then $default_P(p[s])$.
(This case contains the one that there is no rule of the form $body \rhd p[s]$ without any rule of the form $body' \rhd \sim p[s]$, where $default_P(p[s])$.)
(b) If there is a rule $body \rhd \sim p[s]$ such that $imple_P[q[s'])$ with $commu(s',s,q[s'])$ for any $q[s']$ of $body$ and $default_P(r[s''])$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$, then $default_P(p[s])$.

These predicates are concerned with a fixed point model of $P$, where they are made use of for analysis of strategy construction with communication and are related to strategy implementability.

**Proposition 2.** *Assume a program $P$ over the set $A$. Let $(I,J)$ be a pair defined by the $imple_P$ and $default_P$ predicates in the manner:*

$$I = \{p[s] \mid imple_P(p[s])\} \text{ and}$$
$$J = \{p[s] \mid default_P(p[s])\}.$$

*Then $Tr_P(I,J) = (I,J)$, that is, $(I,J)$ is a fixed point of $Tr_P$.*

*Proof.* Let $Tr_P(I,J) = (I_0,J_0)$.
(1) We prove by induction that $(I_0,J_0) \subseteq_c (I,J)$, i.e. $I_0 \subseteq I$ and $J_0 \subseteq J$.
(i) If $p[s]$ is in $I_0$, then there is a rule $body \rhd p[s]$ such that $q[s']$ is in $I$ with $commu(s',s,q[s'])$ for any $q[s']$ of $body$, and $r[s'']$ is in $J$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$. This reason is applied to the case that $body$ is the empty set. By definition of $(I,J)$, $imple_P(q[s'])$, for $q[s']$ to be in $I$, and $default_P(r[s''])$, for $r[s'']$ to be in $J$. It follows from the inductive definition that $imple_P(p[s])$. Thus, $p[s]$ is in $I$, i.e., $I_0 \subseteq I$.
(ii) Assume that $p[s]$ is in $J_0$. Then, there are two cases: • For any rule $body \rhd p[s]$, $q[s']$ is in $J$ with $commu(s',s,q[s'])$ for some $q[s']$ of $body$, or $r[s''])$ is in $I$ with $commu(s'',s,r[s''])$ for some $\sim r[s'']$ of $body$. That is, there is $default_P(q[s'])$, or $imple_P(r[s''])$. It follows that $default_P(p[s])$, i.e. $p[s]$ is in $J$.

(This case contains the one that there is no rule of the form $body \rhd p[s]$ without any rule of the form $body' \rhd \sim p[s]$, where $p[s]$ is in $J$.)

• For some rule $body \rhd \sim p[s]$, $q[s']$ is in $I$ with $commu(s',s,q[s'])$ for any $q[s']$ of $body$, and $r[s'']$ is in $J$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$. That is, there are $imple_P(q[s'])$, and $default_P(r[s''])$. It follows that $default_P(p[s])$, i.e. $p[s]$ is in $J$.

In both cases, if $p[s] \in J_0$, then $p[s] \in J$. Thus $J_0 \subseteq J$.

(2) We prove that $(I,J) \subseteq_c (I_0,J_0)$, i.e., $I \subseteq I_0$ and $J \subseteq J_0$.

(i) If $p[s]$ is in $I$, then $imple_P(p[s])$. Thus there is a rule $body \rhd p[s]$ such that $imple_P(q[s'])$ with $comuu(s',s,q[s'])$ for any $q[s']$ of $body$, and $default_P(r[s''])$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$. By definition of $(I,J)$, $q[s']$ is in $I$ with $commu(s',s,q[s'])$, and $r[s'']$ is in $J$ with $commu(s'',s,r[s''])$. By the definition of $Tr_P$, $p[s]$ is in $I_0$, obtained by applying of $Tr_P$ to the pair $(I,J)$. Thus, if $p[s] \in I$ then $p[s] \in I_0$ (i.e. $I \subseteq I_0$).

(ii) If $p[s]$ is in $J$, then $default_P(p[s])$. There are two cases:

• For any rule $body \rhd p[s]$ such that $default_P(q[s'])$ with $commu(s',s,q[s'])$ for some $q[s']$ of $body$, or $imple_P(r[s''])$ with $commu(s'',s,r[s''])$ for some $\sim r[s'']$ of $body$. By the definition of $(I,J)$, $q[r']$ is in $J$ with $commu(s',s,q[s'])$, or $r[s'']$ is in $I$ with $commu(s'',s,r[s''])$. By the definition of $Tr_P$, $p[s]$ is in $J_0$, obtained by applying of $Tr_P$ to the pair $(I,J)$.

(This case contains the one that there is no rule of the form $body \rhd p[s]$ without any rule of the form $body' \rhd \sim p[s]$, where $p[s]$ is in $J_0$.)

• For some rule $body \rhd \sim p[s]$ such that $imple_P(q[s'])$ with $commu(s',s,q[s'])$ for any $q[s']$ of $body$, and $default_P(r[s''])$ with $commu(s'',s,r[s''])$ for any $\sim r[s'']$ of $body$. By the definition of $(I,J)$, $q[r']$ is in $I$ with $commu(s',s,q[s'])$, and $r[s'']$ is in $J$ with $commu(s'',s,r[s''])$. By the definition of $Tr_P$, $p[s]$ is in $J_0$, obtained by applying of $Tr_P$ to the pair $(I,J)$.

In both cases, if $p[s] \in J$ then $p[s] \in J_0$. Thus $J \subseteq J_0$. □

With Propositions 1 and 2, we can have the meaning that the predicates may be related to a model of the program.

**Proposition 3.** *Assume that for the program P, a pair $(I,J)$ is defined such that*

$$I = \{p[s] \mid imple_P(p[s])\}, \text{ and}$$
$$J = \{p[s] \mid default_P(p[s])\}.$$

*If $I \cap J = \emptyset$, then $(I,J)$ is a model of P.*

## 4.2 Application to Logical Database and Related Works

By regarding strategies (called by name) as data, the program $P$ (in coordination of strategy with communication) describe as above is applicable to logical database where rules are logically described:

(i) As regards negatives, *prohibition* (strong negation *not*) can be used, as well as default $\sim$ for *impermisisibility*, in 3-valued domain.

(ii) The predicates $imple_P$(-) and $default_P$(-) are extended to the predicates for queries to logical database.

Formally we have database *Database* with communication *Commu* (of Section 3), in BNF:

$$Literal ::= p[s] \mid \sim p[s] \mid not\ p[s]$$
$$Head ::= p[s] \mid \sim p[s]$$
$$Body ::= \{\ \} \mid \{Literal\} \cup Body$$
$$Rule ::= Body \rhd Head$$
$$database ::= \emptyset \mid \{Rule\} \cup database$$
$$Database ::= database \cup Commu$$

Note that *Head* does not contain strong negation *not*, and that *Condition* for *Strategy* is also assumed for *database* where the rules of *Rule* have restrictions such that at most one any complementary pair $(p[s], \sim p[s])$ occurs at heads.

About related works on logical frameworks possibly for cognitive managements and for defeasible logic (Governatori et al., 2004), we have examined concepts and ideas. On the one side, quantifications for proposition variables are studied. On the other hand, modal operators are invented with theoretical basis and applicable aspects. They may be relevant to ontology views on structural and knowledgeable analyses of procedures and reasoning:

(i) Applying 3-valued models to the Heyting algebra, we made the papers on modal mu-calculus, a language system and reference data abstraction (S. Yamasaki et al. in *COMPLEXIS* 2020 and *DATA* 2020). The algebraic expressions of those papers are evaluated in 3-valued domain, to possibly represent an infinite conjunctive form of propositional logic, where the form of algebraic expressions should supposedly take conditions similar to *Condition* of this paper, and the prefixpoint models must be a little more restricted than those described there.

(ii) There is a paper presenting second-order propositional frameworks, with epistemic and intuitionistic logic (P. Kremer, 2018). It may be relevant to the extension of this paper with logical expressions to be quantified.

(iii) For an extension of propositional modal logic without quantification (whose transition system is

captured as abstract state machinery), the paper (Fitting, 2002) introduces relations and terms with scoping mechanism by lambda abstraction.
(iv) Based on beliefs and intentions, modal operations have been applied to mental states (Dragoni et al., 1985). The paper (Beddor and Goldstein, 2018) presents the belief predicate with the credence function of agents, concerning epistemic contradictions. The contradictions of complexity may be avoided by grades of such a function.

# 5 CONCLUSION

This paper refines 3-valued model theory of logical expressions from structural aspects of strategic constructions. There is a complexity in that the state constraint strategies inductively form a strategy possibly assigned to another state. Implementability of a strategy is supposedly indebted to implementabilities of the strategies as components to the primary strategy. Non-implementability is denoted with default negation, as well as the undefined for implementability for strategies. The structure of strategy constructions is so far examined with respect to implementability evaluation. A communicative predicate on the set of states is assumed in a simpler manner, such that transferrability of distributed strategies may be virtually supposed. The structure is regarded as relevant to interests of cognitive management complexity caused by combination of computing mechanism with communication between distributed states. With respect to structure of strategies for computing implementability backed by communication facilities, the main results are listed up:

(i) A general form of logical expressions for strategy constructions is formulated with distributions to states, which may be also modeling of distributed logical formulas with default negation.
(ii) 3-valued model theory of such expressions is given in terms of fixed point of the mapping (corresponding to a transformation) associated with expressions attached to states, with respect to predicates for communication. The model denotes implementability of constructed strategies.
(iii) From the predicate and derivation views, we can have the implementable and default predicates for expression evaluations, in accordance with the success and failure derivations for query in a framework of logical expressions, both of which may be sound to a fixed point model. This is a kind of reasoning with respect to a fixed point model of logical expressions, applicable to logical database.

# REFERENCES


Beddor, B. and Goldstein, S. (2018). Believing epistemic contradictions. *Rev.Symb.Log.*, 11(1):87–114.

Bertolissi, C., Cirstea, H., and Kirchner, C. (2006). Expressing combinatory reduction systems derivations in the rewriting calculus. *Higher-Order.Symbolic.Comput.*, 19(4):345–376.

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

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

Dragoni, A., Giorgini, P., and Serafini, L. (1985). Mental states recognition from communication. *J.Log.Program.*, 2(4):295–312.

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.

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

Governatori, G., Maher, M., Autoniou, G., and Billington, D. (2004). Argumentation semantics for defeasible logic. *J.Log.Comput.*, 14(5):675–702.

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

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

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

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

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

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

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

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

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

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

Reps, T., Schwoon, S., and Somesh, J. (2005). Weighted pushdown systems and their application to interprocedural data flow analysis. *Sci.Comput.Program.*, 58(1-2):206–263.

Rutten, J. (2001). *On Streams and Coinduction*. CWI.

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