経営論集　第29巻第1号　2019年　85–91頁

An Axiomatic System of System FS

Koji KOMATSU

〈**Abstract**〉

Axioms of FS (Frame-Structure) logic are shown in this paper. FS logic is inferior to propositional logic, but superior to predicate, in the expressive power. Predicate logic is too week for representing objects and relations between objects written in natural language. There is no notion of an object in the formula of predicate logic in the first place. Propositional logic, however, is too strong for representing notions in this real world. Individual variables and quantifiers exist in formulas of propositional logic. There is no variable in the sentence of natural language. It is true that there are functional words resemble to quantifiers in natural language. But these words don't bind more than one word. In conclusion, formulas of propositional logic tend to be complex comparing to sentences of natural language. Programming languages for logic programing, such as Prolog, didn't become mainstream, for this reason.

〈**Key words**〉

propositional logic, predicate logic, individual variable, object, natural language

1 Introduction

FS (Frame-Structure) logic is capable of structural knowledge representation and has strict denotational semantics. This paper provides several lines of evidence that the formula of FS logic is simpler than that of predicate logic. The strict semantics of FS was defined in my past paper. In this paper, the axioms of System FS are shown.

The formula of FS has a high affinity to the sentence of natural language. The descriptive power of FS is shown by examples of an inference in this paper. The descriptive power of FS is stronger than propositional logic, however, weaker than predicate logic.

There are several factors that make a mechanical inference difficult in natural language. For example, a high degree of freedom of description, an ambiguity and a dependency on contexts. Natural language is symbolic systems for human being to describe knowledge. In computer science, therefore, predicate logic and its extensions have been frameworks for knowledge

2019年9月30日受理

representation and inference.

Predicate logic is a system which describes knowledge as relations between individual sets. For example, a compound notion “students who like fruits” is interpreted to the product set of two sets. One is the set of “students”. The other is the set of “everyone who likes fruits”. In natural language the clause “who likes fruits” modifies a noun “students”. In other words, “like fruit” can be interpreted as an attribute which “students” have. There is a gap between the description of predicate logic and that of natural language. There is no such notion as an individual variable in natural language. The representational power of predicate logic is strong. A logical expression of predicate logic, however, is too complex because of the existence of an individual variable and quantifier.

There is no internal structure in the proposition of propositional logic. And the proposition of predicate logic is constructed from predicates, functions, individual variables and quantifiers. In contrast to these logics, frame-structure logic is a system that has a structural proposition specified for a sentence of natural language. It has following three features. Compound notations are represented structurally as an object. Relations between notions are represented by attribute pairs and ordering. There is no individual variable in its axiomatic system.

2 Syntax

The syntax of FS logic is defined and some examples are shown. An object relation becomes a logical formula by itself. A conjunction of logical formulas is a logical formula and an implication between two logical formulas also becomes a logical formula.

2.1 Definition of Syntax

2.1.1 Primitive Symbols

(a) Attribute label: l1, l2, …
(b) Object symbol: π, a, b, c, …
(c) Object operator: $\cdot$
(d) Relational operator: $\leq$, *En*
(e) Logical operator: $\Rightarrow$, $\sim$

The object symbol “π” is called a universal object and intuitively means “everything”.

2.1.2 Objects

If t is an object symbol, “t” is an object.

If α and β are objects, “$(\alpha \cdot \beta)$”is an object. This object represents the conjunction of α and β. This object, in particular, is called a compound object.

If t is an object symbol, “$t\ [ap_1, ap_2, \ldots, ap_n]$ $(n \geq 1)$”is an object. This object represents “t

that has [ap_1, ap_2, … ,ap_n] as the attribute": This object, in particular, is called a compound object. Each ap_i is either of the following.

$$ap_i = \begin{cases} Lb \rightarrow \alpha & \cdots \quad (i) \\ Lb \rightarrow \theta & \cdots \quad (ii) \\ Lb \leftarrow \theta & \cdots \quad (iii) \end{cases}$$

Where *Lb* is an attribute label, *α* is an object and *θ* is a set of objects (may be an empty set). The ap_i of type (i) is called a single-valued existential attribute pair, (ii) called a multiple-valued existential attribute pair and (iii) is called a multiple-valued universal attribute pair, or simply, each ap_i is called an attribute pair. An attribute pair means that the value of attribute *Lb* is *α* (or *θ*). [ap_1, ap_2, … , ap_n] is called a list of attribute pairs.

If *Lb* is an attribute label and *α* is an object, then "*Lb* (*α*)" is an object. This object represents "The value of *Lb* which *α* has as an attribute label", and such a *Lb* is called an attribute function.

2.1.3 Object Relations

If *α* and *β* are objects, then the object relation is as follows:

(f) $\alpha \leq \beta$

(g) *En* (α)

Where "$\alpha \leq \beta$" means "*α* is a *β*" and "*En* (α)" means "*α* is an actual being".

2.1.4 Logical Formulas

(a) If *P* is an object relation, then *P* is a logical formula (or simply a formula)

(b) If *P* and *Q* are logical formulas, then ($P \Rightarrow Q$) and $\sim$(*P*) are logical formulas.

Logical operators ∧, ∨, ⇔ may be used, and each can be defined from ⇒ and ∼ in the same way as standard propositional logic.

2.2 Definition of Extended notations

Let *α* and *β* be objects, and […] be a list of attribute pairs.

1. $\alpha / [\ldots] \stackrel{\text{def}}{=} \alpha \leq \pi [\ldots]$
2. $\alpha =_o \beta \stackrel{\text{def}}{=} \alpha \leq \beta \wedge \beta \leq \alpha$
3. $\alpha / \overline{\beta} \stackrel{\text{def}}{=} \alpha \leq \sim En (\alpha \cdot \beta)$
4. $\alpha / \overline{[\ldots]} \stackrel{\text{def}}{=} \alpha \leq \overline{\pi [\ldots]}$

2.3 Examples of an attribute pair

1. [Favorite→fruit] … someone's favorites are only fruits (or a fruit)
2. [Favorite→{fruit}] … fruits (or a fruit) are included in someone's favorites
3. [Favorite←{fruit}] … all fruits are included in someone's favorites

2.4 Examples of an object

1. man [Pet→{dog, cat}] … men who keep dogs (or a dog) and cats (or a cat)
2. (male・student) … school-boys

3. Pet(student) … pets kept by students (or a student)
4. (Pet(male・student)・dog[Sex→male, Color→black]) … black male dogs kept by school-boys

2.5 Examples of an object relation

1. 1. vegan ≤ human[Favorite→{carrot}] … All vegetarians are human who like carrots at least.
2. 2. vegan / [Favorite→{carrot}] … All Vegetarians like carrots at least.
3. vegan / $\overline{[\text{Favorite} \rightarrow \{\text{meat}\}]}$ … All Vegetarians do not like meats.
4. *En* (student・vegan) … Students who are vegan exist (Some students are vegan).
5. school-boy $=_o$ (male・student) … school-boys are equivalent to students whose sex is male.

2.6 Examples of an logical formula

1. (male・student) ≤ student … school-boys are students.
2. school-boy ≤ student ∧ student ≤ human ⇒ school-boy ≤ human … If school-boys are students and students are humans then school-boys are human.

3 Semantics

Semantics of FS logic is defined as follows.

3.1 Definition of semantics

Let α and β be objects and *Lb* be an attribute label. Let a set *O* be a set of all object symbols, a set *N* be that of all attribute labels and a set *U* be a countably non-empty set. An interpretation *I* is a tuple (U, φ, ψ), where φ and ψ are functions defined as follows.

1. $\varphi : O \rightarrow 2^U$ is a function that assigns object symbols to a non-empty subset of *U*, where $\varphi(\pi) = U$.
2. $\psi : N \rightarrow (U \rightarrow 2^U)$ is a function that assigns attribute labels to the function $U \rightarrow 2^U$. For $Lb \in N$, the function $\psi(Lb)$ is simply written as ψ_{Lb}. In general, $\psi_{Lb}(x)(=\psi(Lb)(x))$ represents the value of attribute *Lb* of *x*.

   By $I(U, \varphi, \psi)$, any attribute pair, any list of attribute pairs or any object is assigned to a subset of *U* as follows. And based on these assignments, the truth values of object relations and logical formulas under *I* are defined.
3. Assignment for a single-valued existential attribute pair

$I(Lb \rightarrow \alpha) = \{x \mid \psi_{Lb}(x) \subseteq I(\alpha) \text{ and } \psi_{Lb}(x) \neq \varphi\}$

4. Assignment for a multiple-valued existential attribute pair ($k \geq 1$)
   (a) $I(Lb \rightarrow \{\}) = U$
   (b) $I(Lb \rightarrow \{\alpha\}) = \{x \mid \psi_{Lb}(x) \cap I(\alpha) \neq \varphi\}$
   (c) $I(Lb \rightarrow \{\alpha_1, \ldots, \alpha_k\}) = I(Lb \rightarrow \{\alpha_1\}) \cap \ldots \cap I(Lb \rightarrow \{\alpha_k\})$
5. Assignment for a multiple-valued universal attribute pair ($k \geq 1$)
   (a) $I(Lb \leftarrow \{\}) = U$
   (b) $I(Lb \leftarrow \{\alpha\}) = \{x \mid I(\alpha) \subseteq \psi_{Lb}(x)\}$
   (c) $I(Lb \leftarrow \{\alpha_1, \ldots, \alpha_k\}) = I(Lb \leftarrow \{\alpha_1\}) \cap \ldots \cap I(Lb \leftarrow \{\alpha_k\})$
6. Assignment for lists of attribute pairs ($n \geq 1$)
   $I([ap_1, ap_2, \ldots, ap_n]) = I(ap_1) \cap I(ap_2) \cap \ldots \cap I(ap_n)$
7. Assignment for objects ($n \geq 1$)
   (a) $I(t) = \varphi(t)$
   (b) $I((\alpha \cdot \beta)) = \mathrm{I}(\alpha) \cap \mathrm{I}(\beta)$
   (c) $I(t\,[ap_1, ap_2, \ldots, ap_n]) = I(t) \cap I([ap_1, ap_2, \ldots, ap_n])$
   (d) $I(Lb(\alpha)) = \bigcup_{x \in I(\alpha)} \psi_{Lb}(x)$
8. Truth value assignment for object relations
   (a) $I(\alpha \leq \beta) = \mathbf{T}$ iff $\mathrm{I}(\alpha) \subseteq \mathrm{I}(\beta)$
   (b) $En(\alpha) = \mathbf{T}$ iff $\mathrm{I}(\alpha) \neq \varphi$
   If the above conditions are not satisfied, truth value **F** (false) is assigned.
9. Truth value assignment for logical formulas
   This is defined in the same way as in standard propositional logic.

3.2 Logical consequence

Let *S* be a set of formulas and *P* be a formula. *P* is said to be a logical consequence of *S* iff *P* is **T** for every interpretation under which all formulas in *S* are **T**. When *P* is a logical consequence of *S*, it is written as $S \models P$

4 Axiomatic System

The axioms of FS are the following schema, where *t* is any object symbol, and *α*, *β*, *γ* are objects, *Lb* is any attribute label, […] is any list of attribute pairs, ap_i is any attribute pair. An inference role and a theorem of FS are also defined, where *P*, *Q* are any logical formulas. Next, an example of theorem of FS is shown to clarify that FS can not only represent meanigns but execute inferences in structural form similar to natural language. The soundness and completeness of FS have been proved, however, abbreviated in this paper.

4.1 Axioms

1. axioms of propositional logic
2. $\alpha \leq \pi$
3. $\alpha \leq \alpha$
4. $\alpha \leq \beta \wedge \beta \leq \gamma \Leftrightarrow \alpha \leq \gamma$
5. $\pi \, / \, [Lb \rightarrow \{\}]$
6. $\alpha/[Lb \rightarrow \{\beta\}] \wedge \beta \leq \gamma \Rightarrow \alpha/[Lb \rightarrow \{\gamma\}]$
7. $\alpha \leq (\beta \cdot \gamma) \Leftrightarrow \alpha \leq \beta \wedge \alpha \leq \gamma$
8. $\alpha/[Lb \rightarrow \{\beta_1, \cdots, \beta_n\}] \Leftrightarrow \alpha/[Lb \rightarrow \{\beta_1\}] \wedge \cdots \wedge \alpha/[Lb \rightarrow \{\beta_n\}]$
9. $\alpha \leq t[ap_1, \cdots, ap_n] \Leftrightarrow \alpha \leq t \wedge \alpha/[ap_1] \wedge \cdots \wedge \alpha/[ap_n]$
10. $\alpha \leq \beta \Rightarrow Lb(\alpha) \leq Lb(\beta)$
11. $Lb(\alpha) \leq Lb(\pi[Lb \rightarrow \{\pi\}] \cdot \alpha)$
12. $\alpha/[Lb \rightarrow \{\beta\}] \Rightarrow \alpha/[Lb \rightarrow \{Lb(\alpha) \cdot \beta\}]$
13. $\alpha \leq Lb(\beta) \Rightarrow \alpha \leq Lb(\pi[Lb \rightarrow \{\alpha\}] \cdot \beta)$
14. $\alpha/[Lb \rightarrow \beta] \Rightarrow Lb(\alpha) \leq \beta \wedge \alpha/[Lb \rightarrow \{\beta\}]$
15. $Lb(\alpha) \leq \beta \wedge \alpha/[Lb \rightarrow \{\beta\}] \Rightarrow \alpha/[Lb \rightarrow \beta]$
16. $Lb(\alpha) \leq \beta \Rightarrow (\pi[Lb \rightarrow \{\pi\}] \cdot \alpha)/[Lb \rightarrow \beta]$
17. $En(t)$
18. $\alpha/[Lb \rightarrow \{\beta\}] \wedge En(\alpha) \Rightarrow En(\beta)$
19. $En(Lb(\alpha)) \Rightarrow En(\alpha)$
20. $En(\alpha) \wedge \alpha \leq \beta \Rightarrow En(\beta)$
21. $\sim En(\alpha) \Rightarrow \alpha \leq \beta$
22. $\pi \, / \, [Lb \leftarrow \{\}]$
23. $\alpha \, / \, [Lb \leftarrow \{\beta\}] \wedge \gamma \leq \beta \Rightarrow \alpha \, / \, [Lb \leftarrow \{\gamma\}]$
24. $\alpha/[Lb \leftarrow \{\beta_1, \cdots, \beta_n\}] \Leftrightarrow \alpha/[Lb \leftarrow \{\beta_1\}] \wedge \cdots \wedge \alpha/[Lb \leftarrow \{\beta_n\}]$
25. $En(\alpha) \wedge \alpha \, / \, [Lb \leftarrow \{\beta\}] \Rightarrow \beta \leq Lb(\alpha)$
26. $En(\beta) \wedge \alpha \, / \, [Lb \leftarrow \{\beta\}] \Rightarrow \alpha \, / \, [Lb \rightarrow \{\beta\}]$
27. $\sim En(\alpha) \Rightarrow \pi \, / \, [Lb \leftarrow \{\alpha\}]$

[Inference rule] Q is derived from P and $P \Rightarrow Q$ (modus ponens)

[Theorem] Formulas which is derived from axioms by finite applications of the inference rule or axioms themselves

4.2 Example of an inference

From the following three premises “Vegan like all vegetables”, “Carrot is a vegetable” and “It is not necessary that carrot is a vegetable which is liked by Koji”, the consequence “It is not necessary that Koji is a vegan” can be deduced as follows.

4.3 Example of a theorem

Let premises be the following three formulas.

1. vegan / [Favorite ← {vegitable}]
2. carrot ≤ vegitable
3. ∼ (carrot ≤ Favorite(Koji)・vegitable)

Then the following formula can be derived as a theorem.

$$1 \wedge 2 \wedge 3 \Rightarrow \sim(\text{Koji} \leq \text{vegan})$$

4.4 Example of a proof

The proof of an above theorem is below.

[1] vegan / [Favorite ← {carrot}]・・・premise 1, 2, axiom 1, 24

[2] ∼(koji / [Favorite ← {carrot}])・・・premise 3, axiom 1, 18, 26

[3] ∼(koji ≤ vegan)・・・[1], [2], axiom 1, 4

5 Concluding Remarks

The axioms of FS are shown in this paper. It is true that FS logic is inferior to predicate logic in the expressive power. A significant feature of frame-structure logic is, however, that a compound notion can be represented structurally without individual variables. Owing to this feature, in frame-structure logic a sentence of a natural language is easier to be translated to a logical formula than in predicate logic.

6 References

Komatsu, K. 2016. The Strict Semantics of System FS. *Business Review Faculty of Business Administration Bunkyo Gakuin University* 26 (1):1-11.