Research on a General State Formalization Method from the Perspective of Logic

Siyuan Qiu and Jianfeng Xu [*]

Posted Date: 13 August 2025

doi: 10.20944/preprints202508.0939.v1

Keywords: objective information theory; logical systems; states; formal methods

Article

Research on a General State Formalization Method from the Perspective of Logic

Siyuan Qiu [1] and Jianfeng Xu [1,2,*]

[1] Shanghai Jiao Tong University, Shanghai, 200030, China
[2] Koguan School of Law, China Institute for Smart Justice, School of Computer Science
[*] Correspondence: xujf@sjtu.edu.cn

Abstract

As the world rapidly develops, information, as a vital resource, remains a subject of debate, with its definition and nature still being debated. To address this issue, Objective Information Theory proposes a set of axioms that rigorously define information. This paper aims to construct a formal system of mathematical logic using first-order and higher-order logic. Using well-formed formulas, it formalizes states and demonstrates that nearly all structures and states in various fields can be expressed. Finally, this paper proposes a universal state representation method, which improves the definition of state in Objective Information Theory and builds a bridge for the exchange and research of states across various fields.

Keywords: objective information theory; logical systems; states; formal methods

1. Introduction

Matter, energy and information are the three basic elements of nature.[1] In the current era of rapid development of information technology and artificial intelligence, information, as a bridge connecting the material world and the cognitive world, is becoming more and more important than ever. However, although people are talking about the "information revolution" with great interest and a large number of scholars are constantly devoting themselves to the research of informatics, the progress of information theory is not optimistic. The question of the nature of information has always troubled us[2–4].The concept of informatics, which takes information as its object, was proposed as early as the early 1990s, but has made little progress in the past 30 years and has failed to have a significant impact on the development of information technology and systems[5].The reasons for this are: first, there is a lack of a rigorous, universal definition of information; second, the phenomena encompassed by "information" are incredibly diverse, making it difficult to arrive at a comprehensive definition and establish a unified information theory; third, using only information quantity and information entropy as information metrics is too simplistic and ill-suited to the rich functionality of existing information systems; and fourth, current "information theory" is essentially still "communication coding theory." While it covers several important aspects of information encoding, transmission, and storage, it largely ignores aspects of information understanding and processing. Therefore, the development of a comprehensive theoretical framework for informatics is imperative.

Starting from Wiener's famous statement that "information is information, not matter, not energy", Chinese scholar Xu Jianfeng regards information as the third basic category in the objective world, alongside matter and energy. He established a set of basic information postulates, mathematical definitions, a six-tuple model, five properties and 11 types of measurement systems based on the "reflective information view", forming the theoretical system of Objective Information Theory (OIT). He proved that several previously unrelated classical information principles and formulas can be uniformly expressed through OIT[6–8].On this basis, the dynamic configuration and metric efficiency distribution of information systems were established, forming a theoretical framework for information system dynamics. In short, objective information theory, based on people's common understanding of information, defines the information ontology set o, the ontology occurrence time set T_h, the ontology state set $S_o(o, T_h)$, the objective carrier set c, the carrier reflection time set T_m, and the carrier reflection set $S_c(c, T_m)$. Information I is the enabling mapping from the ontology state set $S_o(o, T_h)$ to the carrier reflection set $S_c(c, T_m)$.

However, objective information theory lacks a rigorous mathematical definition of state, and its theoretical foundation remains incomplete. To address this issue, this paper aims to integrate the research findings of classical information theory, supplement the definition and model of information, and construct a formal representation of the object state, further clarifying its composition. Starting from the perspective of mathematical logic, this paper rigorously defines first-order and higher-order predicate logic systems, connects state with the interpretation of sets of well-formed formulas, and develops a comprehensive formal representation of state. This approach fills the existing theoretical gaps in OIT theory and further enhances research on the nature of information. Furthermore, from the perspectives of mathematics, economics, sociology, computer science, and natural linguistics, this paper abstracts typical phenomena from various disciplines, providing a universal symbolic and formal representation of the state within them.

Leveraging this universal formalization, first-order and higher-order logic become a bridge connecting the various fields. This reveals a profound unity in the development of modern science and technology: while different disciplines superficially deal with vastly different objects and phenomena, at a deeper level, they can all be uniformly expressed and mutually transformed through the language of logic.

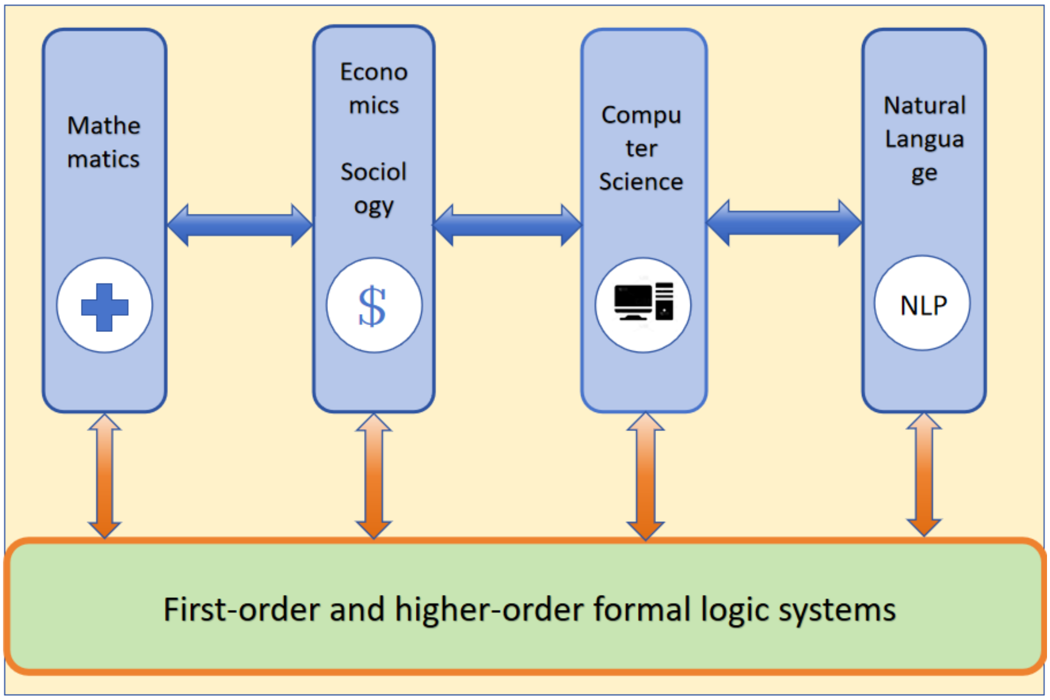

Figure 1. The logical system has become a bridge for communication between various fields and is the most universal language.

2. Formal Expression of State

The information postulate emphasizes that the state of the ontology can be mapped to the state of the carrier. Therefore, establishing a formal representation of the ontology and carrier states is particularly important. Mathematical logic, based on axiomatic systems and symbolic language, enables rigorous and unambiguous characterization of concepts, propositions, and their reasoning. It avoids the ambiguity and uncertainty of natural language expressions and provides a precise descriptive tool for scientific research. Professor Xu Jianfeng of Shanghai Jiao Tong University provided me with a reference for this approach, and based on his work, I conducted further research and expansion. Let's first limit the object language to first-order languages.

2.1. First-Order Formal System Definition

First-order predicate logic language has become a core tool for formal modeling and automatic reasoning in fields such as mathematics, information science, and artificial intelligence due to its strong expressiveness, clear structure, rigorous reasoning, good computability, and strong versatility[9].First, we define the most basic symbols in the first-order formal system:

Definition 1 (Symbols in $\mathcal{L}^{(1)}$)**.** *$\mathcal{L}^{(1)}$ contains the following symbols:*

- *First-order variables:*$x_1^{(1)}, x_2^{(1)}, \ldots$
- *First-order constants:*$a_1^{(1)}, a_2^{(1)}, \ldots$
- *First-order function symbols:*$f_1^{(1)1}, f_2^{(1)1}, \ldots, f_1^{(1)2}, f_2^{(1)2}, \ldots$
- *brackets:*(,)
- *First-order predicate symbols:*$A_1^{(1)1}, A_2^{(1)1}, \ldots, A_1^{(1)2}, A_2^{(1)2}, \ldots$
- *Logical connectives:*$\sim$ *(Negation)*,$\rightarrow$ *(Implication)*
- *Quantifiers:*$\forall$ *(Universal quantifier)*

Based on logical theory, symbols such as conjunction $\wedge$, disjunction $\vee$, if and only if $\leftrightarrow$, and existence $\exists$ can be naturally introduced. In particular, some common mathematical symbols, such as $= and \in$, are also considered universal predicate symbols and will not be discussed further.

Terms in a language are similar to nouns or noun phrases in a natural language, but terms and nouns (phrases) are not exactly the same. The main difference is that terms contain variables and are "compound" items constructed using variables.

Definition 2 (Terms in $\mathcal{L}^{(1)}$)**.** *The terms in $\mathcal{L}^{(1)}$ are generated as follows:*

(1) *Variables and constants are terms:*

(2) *If $f_i^{(1)n}$ $(n > 0, i > 0)$ is a function symbol in $\mathcal{L}^{(1)}$ and $u_1, \ldots, u_n$ is a term in $\mathcal{L}^{(1)}$, then $f_i^{(1)n}(u_1, \ldots, u_n)$ is also a term in $\mathcal{L}^{(1)}$.*

Next, we define atomic formulas. Atomic formulas are the most basic formulas in the language.

Definition 3 (Atomic formulas in $\mathcal{L}^{(1)}$)**.** *If $A_i^{(1)n}$ $(n > 0, i > 0)$ is a predicate symbol in $\mathcal{L}^{(1)}$ and $u_1, \ldots, u_n$ is a term in $\mathcal{L}^{(1)}$, then $A_i^{(1)n}(u_1, \ldots, u_n)$ is an atomic formula in $\mathcal{L}^{(1)}$.*

Definition 4 (Well-formed formulas in $\mathcal{L}^{(1)}$)**.** *The well-formed formula in $\mathcal{L}^{(1)}$ is defined as follows:*

(1) *Each atomic formula is a well-formed formula in $\mathcal{L}^{(1)}$;*

(2) *If $\mathcal{A}$ and $\mathcal{B}$ are well-formed formulas in $\mathcal{L}^{(1)}$, then $\sim \mathcal{A}$ and $\mathcal{A} \rightarrow \mathcal{B}$ are both well-formed formulas in $\mathcal{L}^{(1)}$;*

(3) *If $\mathcal{A}$ is a well-formed formula in $\mathcal{L}^{(1)}$ and u is a variable or function symbol in $\mathcal{L}^{(1)}$, then $(\forall u)\mathcal{A}$ is a well-formed formula in $\mathcal{L}^{(1)}$.*

2.2. Recursive Definition of Higher-Order Formal Systems

Considering that first-order logic still has many shortcomings in terms of quantified objects, recursive induction, and other issues, such as its inability to fully express the concept of a set and its lack of direct characterization of higher-order properties, we need to expand the characterization capabilities of logical systems.Higher-order predicate logic (HOL) is an extension of first-order predicate logic. It allows quantification over predicates, functions, and even predicates about predicates. It offers greater expressive power, can formalize complex semantics in natural language and mathematics, and supports a richer set of logical tools and theoretical frameworks. Next, we will define higher-order formal systems.

Assume that for $1, 2, \ldots, k-1$, the corresponding formal systems $\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(k-1)}$ have been defined. Then the symbols in the k-order formal system $\mathcal{L}^{(k)}$ can be recursively defined[10].

Definition 5 (Symbols in $\mathcal{L}^{(k)}$)**.** *The symbols in $\mathcal{L}^{(k)}$ include:*

- *All symbols of $\mathcal{L}^{(k-1)}$;*
- *k-order variables:$x_1^{(k)}, x_2^{(k)}, \ldots$*
- *k-order constants:$a_1^{(k)}, a_2^{(k)}, \ldots$*
- *k-order predicate variables:$P_1^{(k)1}, P_2^{(k)1}, \ldots, P_1^{(k)2}, P_2^{(k)2}, \ldots$*
- *k order function symbols:$f_1^{(k)1}, f_2^{(k)1}, \ldots, f_1^{(k)2}, f_2^{(k)2}, \ldots$*
- *k-order predicate symbols:$A_1^{(k)1}, A_2^{(k)1}, \ldots, A_1^{(k)2}, A_2^{(k)2}, \ldots$*

Similarly, we can recursively define the terms in $\mathcal{L}^{(k)}$, atomic formulas and well-formed formulas.

Definition 6 (Terms in $\mathcal{L}^{(k)}$)**.** *The terms in $\mathcal{L}^{(k)}$ are defined as follows:*

(1) All terms of $\mathcal{L}^{(k-1)}$;

(2) If $f_i^{(k)n}$ $(n > 0, i > 0)$ is a k-order function symbol in $\mathcal{L}^{(k)}$ and $u_1, \ldots, u_n$ are variables, constants, or functions in $\mathcal{L}^{(k)}$, then $f_i^{(k)n}(u_1, \ldots, u_n)$ is a k-order term in $\mathcal{L}^{(k)}$.

Definition 7 (Atomic formulas in $\mathcal{L}^{(k)}$)**.** *The atomic formula in $\mathcal{L}^{(k)}$ is defined as follows:*

(1) All atomic formulas of $\mathcal{L}^{(k-1)}$;

(2) If $A_i^{(k)n}$ $(n > 0, i > 0)$ is a predicate symbol of order k in $\mathcal{L}^{(k)}$ and $u_1, \ldots, u_n$ are terms in $\mathcal{L}^{(k)}$, then $A_i^{(k)n}(u_1, \ldots, u_n)$ is an atomic formula of order k in $\mathcal{L}^{(k)}$.

Definition 8 (Well-formed formula in $\mathcal{L}^{(k)}$)**.** *The well-formed formula in $\mathcal{L}^{(k)}$ is defined as follows:*

(1) All well-formed formulas for $\mathcal{L}^{(k-1)}$;

(2) If $\mathcal{A}$ and $\mathcal{B}$ are well-formed formulas in $\mathcal{L}^{(k)}$, then $\sim \mathcal{A}$ and $\mathcal{A} \to \mathcal{B}$ are both well-formed formulas in $\mathcal{L}^{(k)}$;

(3) If $\mathcal{A}$ is a well-formed formula in $\mathcal{L}^{(k)}$ and u is an argument or function symbol in $\mathcal{L}^{(k)}$, then $(\forall u)\mathcal{A}$ is a well-formed formula in $\mathcal{L}^{(k)}$.

2.3. Interpretation of Formal Systems

Next, we can define the interpretation of the formal system.

Definition 9 (Interpretation of formal systems)**.** *An interpretation E of the formal system $\mathcal{L}$ is a two-tuple $E = \langle D_E, J \rangle$. Where:*

- *The domain D_E is a non-empty set that contains the value range of all elements in $\mathcal{L}$, including individuals, properties, relations, and functions.*
- *The interpretation function J is a mapping that maps symbols in $\mathcal{L}$ to concrete semantics in the domain D_E and is defined as follows:*
  - *Interpretation of constants and variables: Each constant $a_i^{(k)}$ is interpreted as an element in D_E, i.e., $J(a_i^{(k)}) \in D_E$; each variable $x_i^{(k)}$ is interpreted as an element in D_E, i.e., $J(x_i^{(k)}) \in D_E$.*
  - *Interpretation of function symbols: Each function symbol $f_i^{(k)n}$ is interpreted as a mapping from D_E^n to D_E, that is, $J(f_i^{(k)n}) : D_E^n \to D_E$.*
  - *Interpretation of predicate symbols: Each predicate symbol $A_i^{(k)n}$ is interpreted as a mapping from D_E^n to $\{True, False\}$, i.e., $J(A_i^{(k)n}) : D_E^n \to \{True, False\}$.*

- *Interpretation of the terms:*

$$J(u)=\begin{cases} J(a_i^{(k)}), & \text{if } u \text{ is constant } a_i^{(k)} \\ J(x_i^{(k)}), & \text{if } u \text{ is variable } x_i^{(k)} \\ J(f_i^{(k)n})(J(u_1),\ldots,J(u_n)), & \text{if } u = f_i^{(k)n}(u_1,\ldots,u_n) \end{cases}$$

- *Interpretation of atomic formula:*

$$if A = A_i^{(k)n}(u_1,\ldots,u_n), then J(A) = J(A_i^{(k)n})(J(u_1),\ldots,J(u_n)).$$

- *Interpretation of logical connectives:*

$$J(\sim \mathcal{A}) = True \iff J(\mathcal{A}) = False$$

$$J(\mathcal{A} \to \mathcal{B}) = True \iff (J(\mathcal{A}) \to J(\mathcal{B})) = True$$

- *Interpretation of quantifiers: If $\mathcal{A} = (\forall u)\mathcal{B}$, where u is a variable or function symbol, then*

$$J(\mathcal{A}) = True \iff \forall d \in D_E, \text{when } u \text{ is interpreted as } d,\ J(\mathcal{B}) = True$$

The interpretation of formal systems transforms the abstract syntax of formal logic into concrete semantics, serving as a crucial link between the "symbolic world" and the "real world." It not only renders logical language meaningful but also provides a theoretical foundation for correct reasoning, modeling, verification, and automated applications. It is an essential concept in mathematical logic and information science.

2.4. Axiom System for Logical Expression of Ontology Components Under State Decomposition

Let X denote a set of objects, T denote a set of time points or intervals, and L denote a higher-order formal system. We propose the following four axioms[11].

1. **Parameter Reference Axiom** : Every object $x \in X$, every moment or period $t \in T$, and every function $f \in F$ is represented by a unique constant or term c_x, c_t, c_f in L.
2. **Property Expressibility Axiom** : The properties, form, value, relationship and other attributes of a set of objects in the entire domain can be expressed through functions and predicates in the formal system.
3. **Logical Combination and the Closure Axiom**
   The generation rules of the state space $\mathcal{S}$ are limited to the following logical operations:
   - **Implication**:If $S_1, S_2 \in \mathcal{S}$,then it implies that $S_1 \to S_2$ is also a state of $\mathcal{S}$;
   - **Negation**:If $S \in \mathcal{S}$, then $\neg S \in \mathcal{S}$;
   - **Quantification**: For any state predicate $S(x)$, $\forall x\, S(x)$ and $\exists x\, S(x)$ also belong to $\mathcal{S}$.

   In other words, $\mathcal{S}$ is closed with respect to the above logical operations and only allows new states to be generated through a finite number of implications, negations, and quantifications.
4. **Axiom of temporal causality**:When any attribute, relationship, or state is established at a certain moment, its change or evolution at subsequent moments can be described by the formula in L.

Theorem 1. *If for any $x \in X$, $t \in T$, there exists at least one attribute, relation, or property that can be expressed by L, and satisfies the axioms of parameter reference, attribute expressibility, logical combination closure, and temporal causality, then all states of any object x at any time t can be uniquely characterized by a set of well-formed formulas $\varphi_{S(x,t)}$ in L.*

Proof of Theorem1. According to the parameter reference axiom, the object x, time t, and the function f involved in the set can all be represented by unique terms c_x, c_t, c_f in L.

Next, according to the property expressibility axiom, the various properties of the state set $S(x,t)$ can be expressed using functions and predicates in the formal system.

By definition, all expressions in the above state sets are atomic formulas in L. By logical combination and the closure axiom, any complex ontological state G can be recursively constructed from the base state by applying a finite number of generation rules expressible in L. Each generation rule is uniquely described by a well-formed formula and inference rule in L. Therefore, for any object x and time t, all its ontological states $S(x,t)$ can be uniquely mapped and characterized in L by the corresponding set of formulas $\varphi_{S(x,t)}$, where $\varphi_{S(x,t)}$ is recursively generated from the atomic formulas using logical rules.

In terms of uniqueness, the construction of $S(x,t)$ depends solely on x, t, and the set of ontological components. The representation of all predicates, functions, and parameters in L is uniquely determined by the axiomatic system. Therefore, $\varphi_{S(x,t)}$ uniquely corresponds to $S(x,t)$ within L. If $\varphi', \varphi'' \in L$ both characterize $S(x,t)$, then by the logical equivalence relation in L, $\varphi' \equiv \varphi_{S(x,t)} \equiv \varphi''$, guaranteeing uniqueness.

Furthermore, the temporal causality axiom states that any time t can be expressed by a recursive or evolutionary formula in L. Specifically, for any $x \in X$, there exists a formula $\psi(x,t',x,t)$ in L such that $S(x,t')$ is uniquely determined by $S(x,t)$ and related laws. Thus, any dynamic evolution of a system can be recursively expressed by a chain of well-formed formulas in L, and the history and future of its state can be reduced to the logical deduction of a set of formulas.

In summary, $S(x,t)$ can always be rigorously characterized by a unique set of well-formulated formulas in L under interpretation, and this expression holds true for any dynamic evolution. The theorem is proved. □

2.5. The State of an Object at a Specific Time

Then, according to the theorem, we give the definition of state:

Definition 10 (state)**.** *The state $S(x,t)$ of a set of objects x at a particular time set t is an interpretation of a set of well-formed formulas in the formal system $\mathcal{L}$ on the universe $x \times t$. The specific properties of x and t, as well as the choice of formula set and the definition of the interpretation, are determined by the specific application scenario.*

Theorem 1 and the definition of state demonstrate that state can be well described using first-order and higher-order logic. The formal representation of state is not only a technical tool but also a fundamental way for humans to understand and transform the world. It transforms intuitive concepts into precise mathematical objects, enabling rigorous reasoning and systematic analysis. As science and technology become increasingly complex, this formalization capability will continue to be a vital force driving the development of informatics and the progress of human civilization.

3. Mathematical Field State Expression

Mathematics, as a fundamental discipline, encompasses a wide range of branches and a vast system. Fundamental fields such as number theory focus on the properties of integers; algebra encompasses linear algebra (vector spaces and matrices), abstract algebra (groups, rings, and fields), and polynomial theory; and geometry includes Euclidean geometry and differential geometry (manifolds and curvature). Applied mathematics encompasses topology, probability theory and statistics (such as stochastic processes, Bayesian statistics, and the foundations of machine learning), and computational mathematics (numerical analysis, algorithm design, and scientific computing). Furthermore, new problems continue to emerge in discrete mathematics, mathematical physics, logic, and set theory.

From elementary arithmetic to cutting-edge research, mathematics demonstrates a progression in depth and abstraction, with numerous fields intersecting and integrating. Theorems, propositions, and formulas within each branch can be viewed as characterizing the “state” of certain mathematical objects. Broadly speaking, the state of mathematical objects is a core concept for understanding the dynamics, contextual dependence, and inherent connections of mathematical structures. Studying mathematical states not only helps focus on key properties and ignore minor details, but also helps grasp the essence of a problem, forming a crucial foundation for the development of mathematical theory.

3.1. Formalization of Finite Mathematical Structures

Finite structures are the foundation of discrete mathematics. Problems such as finite sets and their subsets, and the connectivity, coloring, and matching of graphs in finite graph theory are all inseparable from the study of finite structures. In addition, many "infinite" mathematical concepts originate from the generalization of finite structures[12–14].

Finite mathematical structures are not only an essential component of mathematics but also fundamental tools for understanding the complex world, solving practical problems, and advancing science and technology. In a sense, finite structures are "real mathematics"—both theoretically elegant and practically useful. Here, we provide a rigorous proof that finite mathematical structures can be formalized in a first-order manner.

Definition 11 (Finite structure). *Let $\mathfrak{A} = \langle A, R_1^{\mathfrak{A}}, \ldots, R_m^{\mathfrak{A}}, f_1^{\mathfrak{A}}, \ldots, f_n^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \ldots, c_k^{\mathfrak{A}} \rangle$ be a finite structure, where: A is a finite set, $|A| = N$, $R_i^{\mathfrak{A}} \subseteq A^{a_i}$ is an a_i meta-relation, $f_j^{\mathfrak{A}} : A^{b_j} \to A$ is a b_j meta-function, and $c_l^{\mathfrak{A}} \in A$ It's constant.*

Theorem 2 (First-order complete characterization of finite structures). *If $\mathfrak{A}$ is a finite structure, then there exists a first-order language L and a set of L-sentences Γ such that for any L-structure $\mathfrak{B}$:*

$$\mathfrak{B} \models \Gamma \textit{ if and only if } \mathfrak{B} \cong \mathfrak{A} \tag{1}$$

Proof of Theorem2. The definition L includes:

- Relation symbols: $R_1, \ldots, R_m$ (with arity $a_1, \ldots, a_m$ respectively)
- Function symbols: $f_1, \ldots, f_n$ (with arity $b_1, \ldots, b_n$ respectively)
- Constant symbols: $c_1, \ldots, c_k$
- Individual constants: $d_1, \ldots, d_N$ (corresponding to each element in A)

Let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, construct the following statement:

(Γ_1) domain restriction statement:

$$\forall x(x = d_1 \vee x = d_2 \vee \cdots \vee x = d_N) \tag{2}$$

(Γ_2) element-wise distinction statement:

$$d_i \neq d_j \quad \text{(for all } 1 \leq i < j \leq N) \tag{3}$$

(Γ_3) relation characterization statement:
For each relation symbol R_i and each tuple $(\alpha_{j_1}, \ldots, \alpha_{j_{a_i}}) \in A^{a_i}$:

$$\begin{cases} R_i(d_{j_1}, \ldots, d_{j_{a_i}}) & \text{If } (\alpha_{j_1}, \ldots, \alpha_{j_{a_i}}) \in R_i^{\mathfrak{A}} \\ \neg R_i(d_{j_1}, \ldots, d_{j_{a_i}}) & \text{If } (\alpha_{j_1}, \ldots, \alpha_{j_{a_i}}) \notin R_i^{\mathfrak{A}} \end{cases} \tag{4}$$

(Γ_4) function characterization statement :
For each function symbol f_j and each tuple $(\alpha_{k_1}, \ldots, \alpha_{k_{b_j}}) \in A^{b_j}$:

$$f_j(d_{k_1}, \ldots, d_{k_{b_j}}) = d_l \tag{5}$$

Where l satisfies $f_j^{\mathfrak{A}}(\alpha_{k_1}, \ldots, \alpha_{k_{b_j}}) = \alpha_l$

doi:10.20944/preprints202508.0939.v1

(Γ_5) Constant Characterization Statement:

$$c_i = d_j \quad \text{where } j \text{ satisfies } c_i^{\mathfrak{A}} = \alpha_j \tag{6}$$

We define $\Gamma = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5\}$. Next, we prove two lemmas.

Lemma 1. *If $\mathfrak{B} \models \Gamma$, then $|B| = N$.*

Proof of Lemma1. By (Γ_1), $\forall x \in B, \exists i \in \{1, \ldots, N\}, x = d_i^{\mathfrak{B}}$, so $|B| \leq N$. By (Γ_2), $d_i^{\mathfrak{B}} \neq d_j^{\mathfrak{B}}$ for all $i \neq j$, so $|B| \geq N$. Therefore, $|B| = N$. □

Lemma 2. *If $\mathfrak{B} \models \Gamma$, then the map $h : A \to B$ defined as $h(\alpha_i) = d_i^{\mathfrak{B}}$ is a bijection.*

Proof of Lemma2. This follows directly from Lemma 1 and (Γ_2). □

Next we can prove the consequence of the Theorem2:
($\Rightarrow$) If $\mathfrak{B} \cong \mathfrak{A}$, then $\mathfrak{B} \models \Gamma$:
Let $g : A \to B$ be an isomorphic mapping. Definition of $\mathfrak{B}$:

- $d_i^{\mathfrak{B}} = g(\alpha_i)$
- $R_i^{\mathfrak{B}}, f_j^{\mathfrak{B}}, c_l^{\mathfrak{B}}$ are defined by isomorphic correspondences.

By the definition of isomorphism, $\mathfrak{B}$ satisfies all statements in Γ.
($\Leftarrow$) If $\mathfrak{B} \models \Gamma$, then $\mathfrak{B} \cong \mathfrak{A}$:
By Lemma 2, $h : A \to B$ is defined as $h(\alpha_i) = d_i^{\mathfrak{B}}$, which is a bijection.
Verify that h maintains the relationship:
For any $(\alpha_{j_1}, \ldots, \alpha_{j_{a_i}}) \in A^{a_i}$:

$$\begin{aligned} &(\alpha_{j_1}, \ldots, \alpha_{j_{a_i}}) \in R_i^{\mathfrak{A}} \\ &\iff \mathfrak{B} \models R_i(d_{j_1}, \ldots, d_{j_{a_i}}) \quad (\text{by } (\Gamma_3)) \\ &\iff (d_{j_1}^{\mathfrak{B}}, \ldots, d_{j_{a_i}}^{\mathfrak{B}}) \in R_i^{\mathfrak{B}} \\ &\iff (h(\alpha_{j_1}), \ldots, h(\alpha_{j_{a_i}})) \in R_i^{\mathfrak{B}} \end{aligned} \tag{7}$$

Verify that h holds:
For any $(\alpha_{k_1}, \ldots, \alpha_{k_{b_j}}) \in A^{b_j}$, let $f_j^{\mathfrak{A}}(\alpha_{k_1}, \ldots, \alpha_{k_{b_j}}) = \alpha_l$
From (Γ_4):

$$\begin{aligned} &\mathfrak{B} \models f_j(d_{k_1}, \ldots, d_{k_{b_j}}) = d_l \\ &\iff f_j^{\mathfrak{B}}(d_{k_1}^{\mathfrak{B}}, \ldots, d_{k_{b_j}}^{\mathfrak{B}}) = d_l^{\mathfrak{B}} \\ &\iff f_j^{\mathfrak{B}}(h(\alpha_{k_1}), \ldots, h(\alpha_{k_{b_j}})) = h(\alpha_l) \\ &\iff f_j^{\mathfrak{B}}(h(\alpha_{k_1}), \ldots, h(\alpha_{k_{b_j}})) = h(f_j^{\mathfrak{A}}(\alpha_{k_1}, \ldots, \alpha_{k_{b_j}})) \end{aligned} \tag{8}$$

Verify that h remains constant:This is directly derived from (Γ_5).
Thus, h is an isomorphism, $\mathfrak{B} \cong \mathfrak{A}$. □

We can then draw the following inferences

Corollary 1 (Uniqueness)**.** *The set of axioms Γ uniquely determines $\mathfrak{A}$ in the sense of logical equivalence.*

Corollary 2 (Completeness)**.** *For any first-order property φ on $\mathfrak{A}$, either $\Gamma \models \varphi$ or $\Gamma \models \neg\varphi$.*

Proof of Corollary2. Let φ be any first-order sentence. Since φ is a sentence, either $\mathfrak{A} \models \varphi$ or $\mathfrak{A} \models \neg\varphi$ must hold.

If $\mathfrak{A} \models \varphi$, then by Theorem 1, any structure $\mathfrak{B}$ satisfying Γ is isomorphic to $\mathfrak{A}$, so $\mathfrak{B} \models \varphi$. Therefore, $\Gamma \models \varphi$.

If $\mathfrak{A} \models \neg\varphi$, then similarly, $\Gamma \models \neg\varphi$. □

Corollary 3 (Decidability)**.** *The set $\{\varphi : \Gamma \models \varphi\}$ is decidable.*

Proof of Corollary2. By Corollary 2, for any sentence φ, we can directly verify that $\mathfrak{A} \models \varphi$ on the finite structure $\mathfrak{A}$. If true, then $\Gamma \models \varphi$; otherwise, $\Gamma \models \neg\varphi$. □

3.2. Previous Research on the Formalization of Infinite Structures

Naturally, we will wonder whether all infinite structures, except finite ones, can be completely characterized by a set of first-order logic formulas.

Generally speaking, the answer is no. In fact, according to the research results of Skolem et al., even countable structures cannot be guaranteed to be fully described by first-order logic[15].

Theorem 3 (Skolem)**.** *The standard natural numbers are countable structures that cannot be characterized by first-order theoretical categories.*

We will not give a detailed proof here. The main idea of the proof is to introduce infinite elements through extension theory. Then, we use the compactness theorem to derive a non-standard model and conclude.

Of course, if we expand the tools from first-order logic to higher-order logic, we can expand the characterization capabilities of the logical language[16,17]:

Theorem 4 (Peano)**.** *Under standard second-order semantics, the second-order Peano axioms categorically characterize the structure of natural numbers. That is, if quantification over set variables is allowed, then the sequence of natural numbers can be uniquely characterized by second-order logic.*

This is truly a profound mathematical result, rigorously proven. It not only solves the problem of characterizing natural numbers but also reveals a fundamental property of the expressive power of logical systems—higher-order logic possesses greater expressiveness than first-order logic.

However, despite its greater expressiveness, higher-order logic still cannot represent all countable structures. The boundaries of the logical structures that higher-order logic can represent remain unresolved. This result reflects the fundamental tension between computability and logical expressiveness. Even the most powerful logical systems cannot fully "tame" the complexity of infinite structures. This perhaps reveals a certain irreducible complexity of mathematical reality.

At present, the problem of expressing mathematical structures still depends on the work done by Scott in 1965[18].

Scott first introduced the concept of infinite logic:

Definition 12 (infinite logic$L_{\omega_1\omega}$)**.** *The language $L_{\omega_1\omega}$ is defined by the following rules:*

1. *Contains all atomic formulas of first-order logic*
2. *If $\{\phi_i : i \in I\}$ is a set of formulas and $|I| \leq \aleph_0$, then $\bigwedge_{i\in I} \phi_i$ and $\bigvee_{i\in I} \phi_i$ are also formulas.*
3. *If ϕ is a formula and x is a variable, then $\exists x\phi$ and $\forall x\phi$ are formulas.*
4. *Every formula contains only a finite number of free variables.*

Furthermore, he proposed the crucial isomorphism theorem in the article.

Theorem 5 (Scott's isomorphism theorem, 1965)**.** *Let $\mathfrak{A}$ be a countable structure and $\mathcal{L}$ be a countable language. Then there exists a $L_{\omega_1\omega}$ statement $\phi_{\mathfrak{A}}$ (called a **Scott statement** of $\mathfrak{A}$) such that:*

For any structure $\mathfrak{B}$,

$$\mathfrak{B} \models \phi_{\mathfrak{A}} \iff \mathfrak{B} \cong \mathfrak{A} \tag{9}$$

That is, $\phi_{\mathfrak{A}}$ completely characterizes the structure $\mathfrak{A}$ in an isomorphic sense.

Scott's isomorphism theorem is more than just a technical result; it reveals that infinitely long formulas are a natural tool for dealing with infinite structures, and that abstract existence can be transformed into concrete constructions.

Scott's isomorphism theorem not only solves a specific mathematical problem but, more importantly, opens up a whole new research paradigm, influencing multiple branches of mathematics and still guiding development in related fields today. This makes it one of the most important achievements in mathematical logic of the 20th century.

3.3. Formalization of Conditional Infinite Structures

To address the problem that infinite structures are difficult to characterize using logic, we present and prove a slightly weaker but still highly universal theorem. First, we provide several definitions.

Definition 13 (Relationship maintenance)**.** *Let $\mathfrak{M} = (M, R_1, R_2, \dots, R_k)$ be a structure where R_i is a n_i-ary relation. Let $\{\mathfrak{M}_j = (M_j, R_1^j, R_2^j, \dots, R_k^j)\}_{j\in\mathbb{N}}$ be an approximate sequence.*

***Relationship maintenance** means:*

$$\forall i \in \{1, 2, \dots, k\}, \forall j \in \mathbb{N} : \quad R_i^j = R_i \restriction M_j^{n_i} \tag{10}$$

then

$$R_i^j = \{(a_1, \dots, a_{n_i}) \in M_j^{n_i} : (a_1, \dots, a_{n_i}) \in R_i\} \tag{11}$$

Definition 14 (A precise definition of recursive approximation)**.** *The structure $\mathfrak{M} = (M, R_1, \dots, R_k)$ satisfies **recursive approximation** if and only if:*

There exists a sequence $\{\mathfrak{M}_n\}_{n\in\mathbb{N}}$ where every $\mathfrak{M}_n = (M_n, R_1^n, \dots, R_k^n)$ satisfies:

1. ***Monotonicity:** $\mathfrak{M}_n \subseteq \mathfrak{M}_{n+1} \subseteq \mathfrak{M}$*
2. ***Countability:** $|M_n| = \aleph_0$ for all n*
3. ***Recursion:** There exists a recursive function that computes the Scott statement for each $\mathfrak{M}_n$*
4. ***Density:** $\overline{\bigcup_n M_n} = M$(in appropriate topology)*
5. ***Relationship maintenance:** $R_i^n = R_i \restriction M_n^{ar(R_i)}$ For all i, n, where $ar(R_i)$ denotes the number of elements of the relation R_i*
6. ***Asymptotic uniqueness:** Any two sequences are isomorphic to themselves or to each other after adding a finite number of elements from M.*

Definition 15 (Topology of Scott's statements space)**.** *We define a topology on the space of Scott statements, which makes the convergence exact. We assume that the space is complete under this topology.*

Define $\mathcal{S}$ as the space of all Scott statements. For $\phi, \psi \in \mathcal{S}$, define the distance:

$$d(\phi, \psi) = \sum_{k=1}^{\infty} 2^{-k} \cdot d_k(\phi, \psi) \tag{12}$$

Where:

$$d_k(\phi, \psi) = \frac{|Tp_k(\phi) \triangle Tp_k(\psi)|}{|Tp_k(\phi) \cup Tp_k(\psi)|} \tag{13}$$

Here $Tp_k(\phi)$ denotes the set of k-types that occur in a structure satisfying ϕ, and $Tp_k(\mathfrak{M})$ contains the "complete description" of all possible k-tuples in the structure $\mathfrak{M}$.

Definition 16 (Local finiteness)**.**

$$\forall k, \forall l \in \mathbb{N}, |Tp_k(\phi_l)| = \aleph_0 \tag{14}$$

and

$$\forall k, \forall l \in \mathbb{N}, |Tp_k(\phi_{l+1}) - Tp_k(\phi_l)| < \infty \tag{15}$$

Theorem 6 (Higher-order characterization theorems for recursive approximate structures)**.** *Suppose $\mathfrak{M}, \mathfrak{N}$ is an uncountable structure that satisfies recursive approximation and local finiteness. Then there exists a higher-order logic theory $T_{\mathfrak{M}}$ such that:*

$$\forall \mathfrak{N}(\mathfrak{N} \models T_{\mathfrak{M}} \iff \mathfrak{N} \cong \mathfrak{M}) \tag{16}$$

Here, we assume that higher-order logic can be infinitely quantified, that is, it satisfies the properties of infinite logic. We prove this conclusion step by step. First, we prove that the recursive approximation sequence is inherently unique.

Lemma 3 (Normality of approximate sequences)**.** *Assume $\mathfrak{M}$ satisfies recursive approximation, and $\{\mathfrak{M}_n\}$ and $\{\mathfrak{M}'_n\}$ are two approximate sequences that satisfy the condition. Then there exists an increasing function $h : \mathbb{N} \to \mathbb{N}$ such that:*

$$\mathfrak{M}_n \cong \mathfrak{M}'_{h(n)} \text{ for all } n \tag{17}$$

Proof of Lemma3. By density, monotonicity, and asymptotic uniqueness, for any $\mathfrak{M}_n$, there exists a sufficiently large m such that $\mathfrak{M}_n$ can be embedded in $\mathfrak{M}'_m$.

Vice versa. Combined with countability, we obtain an isomorphism. □

Next we define limit operations and related concepts.

Definition 17 (Limits of structural sequences)**.** *Let $\{\mathfrak{M}_n\}$ be an increasing countable sequence of structures. Definition:*

$$\lim_{n\to\infty} \mathfrak{M}_n = \left(\bigcup_n M_n, \bigcup_n R_1^n, \ldots, \bigcup_n R_k^n\right) \tag{18}$$

If the union of every relation is well-defined in the limit.

Lemma 4 (Existence and uniqueness of limits)**.** *If $\{\mathfrak{M}_n\}$ satisfies the conditions for recursive approximation, then the limit exists and is isomorphic to the original structure $\mathfrak{M}$.*

Proof of Lemma4. Existence: By monotonicity, $\bigcup_n M_n$ is well-defined. The union of relations is well-defined by the relation-preserving property.

Uniqueness: By density, $\bigcup_n M_n$ is dense in M. If $\mathfrak{M}$ has appropriate continuity (implied by the recursive approximation), then it is uniquely determined by the dense substructure. □

Next, we need to verify the convergence of Scott's sequence of statements.

Theorem 7 (Convergence Theorem)**.** *Under the recursive approximation, if the* ***local finiteness condition*** *is additionally satisfied, then the Scott statement sequence $\{\phi_n\}$ converges to ϕ_∞ in the defined topology.*

Proof of Theorem7. For fixed k, consider the sequence:

$$\mathrm{Tp}_k(\phi_1) \subseteq \mathrm{Tp}_k(\phi_2) \subseteq \cdots \subseteq \mathrm{Tp}_k(\phi_{k_m}) \tag{19}$$

For any ϵ, we can choose K so that$\sum_{k=K+1}^{\infty} 2^{-k} < \varepsilon/2$.

Since each inclusion is a subset relation, we can exploit local finiteness. Let N_k exist such that when $m, n \geq N_k$:

$$d_k(\phi_m, \phi_n) < 2^{(k-1)}\epsilon/K \tag{20}$$

So there exists $N = \max\{N_1, N_2, \ldots, N_K\}$ such that when $m, n \geq N$:

$$d_k(\varphi_n, \varphi_{)} < 2^{(k-1)}\epsilon / N \quad \text{for all } k \leq K \tag{21}$$

Thus:

$$\begin{aligned}
d(\varphi_n, \varphi_m) &= \sum_{k=1}^{\infty} 2^{-k} \cdot d_k(\varphi_n, \varphi_m) \\
&= \sum_{k=1}^{K} 2^{-k} \cdot 2^{(k-1)} \varepsilon / K + \sum_{k=K+1}^{\infty} 2^{-k} \cdot d_k(\varphi_n, \varphi_\infty) \\
&\leq \varepsilon/2 + \sum_{k=K+1}^{\infty} 2^{-k} \cdot 1 \\
&< \varepsilon/2 + \varepsilon/2 = \varepsilon
\end{aligned} \tag{22}$$

Thus, we obtain a Cauchy sequence. Based on completeness, we prove that this sequence has a limit. Let's assume that its limit is ϕ_∞. □

Next, we can construct a complete characterization theory:

$$T_{\mathfrak{M}} = T_{\text{approximation}} \cup T_{\text{limit}} \cup T_{\text{unique}} \cup \phi_\infty \tag{23}$$

$$T_{\text{approximation}} = \{\exists \{\mathfrak{M}_n\}_n.\text{Satisfies the recursive approximation condition}\} \tag{24}$$

$$T_{\text{limit}} = \left\{ \mathfrak{M} = \lim_{n \to \infty} \mathfrak{M}_n \right\} \tag{25}$$

$$T_{\text{unique}} = \{\text{The approximation sequence is unique under isomorphism}\} \tag{26}$$

Finally, we prove the Theorem6;

Proof of Theorem6. Assume $\mathfrak{N} \models T_{\mathfrak{M}}$, then $\mathfrak{N}$ has an approximate sequence $\{\mathfrak{N}_n\}$ that satisfies the same conditions.

By $\phi_1, \ldots, \phi_\infty$, the Scott statement for each $\mathfrak{N}_n$ is identical to the corresponding $\mathfrak{M}_n$. By Scott's theorem, $\mathfrak{N}_n \cong \mathfrak{M}_n$ for all n.

Construct an isomorphic sequence $f_n : \mathfrak{M}_n \to \mathfrak{N}_n$. By monotonicity, consistency, and Lemma4, $\{f_n\}$ can be combined into a global isomorphism:

$$f = \bigcup_n f_n : \mathfrak{M} \to \mathfrak{N} \tag{27}$$

Verifying that f is indeed an isomorphism: - Injectivity: by the injectivity and density of each f_n - Surjectivity: by the surjectivity and limit properties of each f_n - Homomorphism: by the preservation of relations. □

At this point, we have rigorously proved the theorem and obtained the most abstract formula for $\mathfrak{M}$, ϕ_∞. Clarifying ϕ_∞ will help researchers understand the logical nature of infinite structures and facilitate deeper research.

3.4. Formalization of Phenomena in Mathematics

Based on the formal characterization of finite and infinite mathematical structures, we finally conclude:

Theorem 8. *Within the first-order and higher-order theoretical framework, almost all mathematical phenomena and mathematical structures can be effectively logically characterized.*

Our theory demonstrates that, within the appropriate framework, nearly all mathematical phenomena—particularly discrete, algebraic, and finitely generated phenomena—can be meaningfully logically characterized. This is an important theoretical achievement that expands our understanding of the extent to which mathematics can be formalized.

However, the richness and complexity of mathematics mean that finding a precise logical expression is difficult. As mathematics develops, modern mathematics presents new challenges. Problems such as the explosion of parameter space, the breakdown of intuition, and insufficient tools make characterizing high-dimensional and abstract problems particularly difficult. This reminds us that mathematics has both a formal side and a side beyond formalism. A perfect logical characterization may be a guiding principle, guiding us to continuously deepen our understanding, but it should not be mistaken for a fully achievable ultimate goal.

4. State Expression in Economics and Sociology

The logical formalization of economic and sociological phenomena is of great significance. First, it provides a scientific theoretical foundation for these disciplines. By using precise mathematical language to eliminate the ambiguity and ambiguity inherent in traditional textual descriptions, it enables rigorous definitions and operational measurement standards for abstract concepts such as market efficiency, social capital, and institutional change. Second, logical formalization builds a bridge for interdisciplinary dialogue, enabling the comparison, integration, and mutual learning of rational choice theory in economics, social network analysis in sociology, and institutional theory in political science within a unified mathematical framework, thus promoting the comprehensive development of the social sciences. Third, it significantly enhances the scientific nature of empirical research. By formalizing theoretical assumptions, it makes research replicable and verifiable, and provides essential mathematical tools for computational social science in the big data era. Finally, at the level of policymaking and social governance, logical formalization enables complex socioeconomic policies to be based on rigorous logical reasoning and mathematical modeling, improving the scientific nature of decision-making and the accuracy of predictions. Overall, logical formalization is driving the transformation of economics and sociology from descriptive disciplines to predictive and explanatory sciences, providing more powerful theoretical tools for understanding and improving human society.

4.1. Logical Characterization in the Field of Economics

The logical characterization of economics is of fundamental significance to the development of the discipline and is also one of the hot issues in research[19–22]. First, logical representation can eliminate ambiguity and vagueness in economic theory. Traditional textual descriptions often allow for multiple interpretations, while logical formalization requires precise definitions of each concept and relationship, forcing theorists to clearly express their assumptions and reasoning. For example, when we say "demand is negatively correlated with price," a logical representation requires us to specify under what conditions, for which goods, and over what timeframe this relationship holds true.

Furthermore, economics deals with complex systems involving multiple agents, multiple levels, and multiple variables, involving interactions between diverse actors such as consumers, businesses, and governments[23]. When a theory becomes complex, natural language descriptions often fail to accurately capture all logical relationships and constraints. Logical representations provide a structured approach to organizing these complex relationships, ensuring the internal consistency and integrity of the theory[24]. For example, when analyzing market equilibrium, we need to simultaneously consider multiple constraints, such as the supply equation, the demand equation, and market-clearing conditions. Logical representations can clearly demonstrate the logical dependencies between these conditions.

At the same time, considering that there are multiple schools and theoretical frameworks within economics, such as neoclassical economics, Keynesianism, institutional economics, etc[25].Different theoretical frameworks utilize different conceptual systems and analytical methods, making academic dialogue difficult. Logical representation provides a unified language for different theories, enabling theoretical comparison, integration, and synthesis. Researchers can more easily identify commonalities and divergences between different theories, promoting the integration and development of theories.

Therefore, the importance of logical characterization in the field of economics is self-evident. Here we first consider the logical characterization of finite economic structures.

Definition 18 (Economic structure). *An economic structure $\mathfrak{S}$ can be represented as follows:*

$$\mathfrak{S} = (A, R_1, R_2, \ldots, F_1, F_2, \ldots, P_1, P_2, \ldots) \tag{28}$$

Where:

- *A is a set of agents (individuals, enterprises, institutions, etc.)*
- *R_i is a relationship (social network, hierarchy, transaction relationship, etc.)*
- *F_j is a function (utility function, production function, decision rule, etc.)*
- *P_k is a process (market mechanism, institutional evolution, information dissemination, etc.)*

Definition 19 (The logical language of economic structure). ***Basic language $\mathcal{L}_{SE}$ Contains:***

Individual constants:

- *$a_1, a_2, \ldots$ represent specific agent individuals, enterprises, organizations, etc.*

Variables:

- *x, y, z represent agent variables, t represents time variables, and s represents state variables*

Predicate symbols:

- *$Agent(x)$: x is an agent*
- *$Transition_{P_k}(s, s')$: represents the transition from state s to state s' under process P_k*
- *$TransitionCondition_k(s, s')$: indicates that the transition from state s to state s' under process P_k satisfies the transition condition.*

Theorem 9 (First-order representability of finite economic structures). *Let $\mathfrak{S} = (A, R_1, R_2, \ldots, F_1, F_2, \ldots, P_1, P_2, \ldots)$ be a finite economic structure, that is:*

1. *$|A| < \infty$ (Finite Agents)*
2. *Every relation R_i and function F_j is defined over a finite field and has corresponding predicate and function representations in the base language.*
3. *The process P_k involves finite states and finite time.*

Then there exists a set of first-order formulas Φ such that:

$$\forall \mathfrak{T}, \mathfrak{T} \models \Phi \iff \mathfrak{T} \cong \mathfrak{S} \tag{29}$$

Proof of Theorem9. **Domain characterization:**

$$\phi_{\text{domain}} = \forall x (Agent(x) \to \bigvee_{i=1}^{|A|} x = a_i) \wedge \bigwedge_{i \neq j} a_i \neq a_j \tag{30}$$

Characterization of relationships: For each relation $R_i \subseteq A^{n_i}$:

$$\phi_{R_i} = \forall x_1 \cdots x_{n_i} (R_i(x_1, \ldots, x_{n_i}) \leftrightarrow \bigvee_{(a_{j_1}, \ldots, a_{j_{n_i}}) \in R_i} (x_1 = a_{j_1} \wedge \cdots \wedge x_{n_i} = a_{j_{n_i}})) \tag{31}$$

Function description: For each function $F_j : A^{m_j} \to D_j$ (where the range D_j is finite):

$$\phi_{F_j} = \forall x_1 \cdots x_{m_j} \exists! y (F_j(x_1, \ldots, x_{m_j}) = y \wedge y \in D_j) \tag{32}$$

Description of the process: For each process P_k, if it involves a finite state transition:

$$\phi_{P_k} = \forall s \forall s' (Transition_{P_k}(s, s') \to \text{TransitionCondition}_k(s, s')) \tag{33}$$

Complete formula:

$$\Phi = \{\phi_{\text{domain}}, \phi_{R_1}, \phi_{R_2}, \ldots, \phi_{F_1}, \phi_{F_2}, \ldots, \phi_{P_1}, \phi_{P_2}, \ldots\} \tag{34}$$

Since all components are finite, every formula is first-order, and Φ completely characterizes the structure $\mathfrak{S}$, we have proved the result. □

From the proof, we can see that structural finiteness has a profound impact on the construction of economic theory. It means that economic models need to pay more attention to boundary conditions, constrained optimization, and finite games. At the same time, finiteness also provides a more realistic foundation for economic analysis, making theoretical predictions more closely aligned with actual economic phenomena.

Given that in the real world, the number of market participants, the variety of resources and commodities, and time and space are all finite, we can further conclude.

Theorem 10. *Within the theoretical framework of first-order and higher-order logic, almost all economic structures can be fully characterized.*

Here, we give an example to verify how logic represents economic phenomena and structures. Lowercase letters represent variables.

Table 1. Predicate Definition.

symbol	definition
$Demand(i, g, p, q)$	The quantity q demanded by consumer i for good g at price p is q
$Consumer(i)$	i is a consumer
$Good(g)$	g is a commodity
$Price(p)$	p is a valid price (non-negative)
$Quantity(q)$	q is a valid quantity (non-negative)
$MaximizesUtility(i, b, constraint)$	Consumer i chooses a bundle of goods b to maximize utility under constraints
$BudgetConstraint(i, p, income)$	Consumer i's budget constraint under price p and income $income$
$Income(i)$	Income of consumer i
$Contains(b, g, q)$	The bundle b contains a quantity q of the good g
$Supply(f, g, p, q)$	The supply of good g by firm f at price p is q
$Firm(f)$	f is an enterprise (production unit)
$ProfitMaximizing(f, v, q_{bundle}, p, w)$	Firm f chooses input v and output q_{bundle} to maximize profit under price p and factor price w^1

Definition 20. ***Definition of Supply and Demand:***

$$\begin{aligned} Demand(i, g, p, q) \leftrightarrow\; & Consumer(i) \wedge Good(g) \wedge Price(p) \wedge Quantity(q) \wedge \\ & \exists b(MaximizesUtility(i, b, BudgetConstraint(i, p, Income(i))) \wedge \\ & Contains(b, g, q)) \end{aligned} \tag{35}$$

$$\begin{aligned} Supply(f, g, p, q) \leftrightarrow\; & Firm(f) \wedge Good(g) \wedge Price(p) \wedge Quantity(q) \wedge \\ & \exists v, q_{bundle}(ProfitMaximizing(f, v, q_{bundle}, p, w) \wedge \\ & Contains(q_{bundle}, g, q)) \end{aligned} \tag{36}$$

4.2. Logical Characterization in the Field of Sociology

The reason why social structures require logical representations is closely related to their inherent complexity and abstractness, and these needs are even more pressing than in economics[26–28].

The need to handle relational complexity is a core motivation for logical representations of social structure. Social structure is essentially composed of a network of multiple relationships between individuals, including kinship, power, economic, and cultural relationships. These relationships often intersect and overlap, forming a multidimensional, complex network. Natural language struggles to accurately describe structural features such as transitivity, symmetry, and hierarchy within such networks. Logical representations can formalize the precise properties of these relationships, such as the transitive relationship "If A is B's superior, and B is C's superior, then A is C's superior."

The need to operationalize abstract concepts is another key factor. Sociology is replete with abstract concepts, such as social status, cultural capital, social cohesion, and institutional legitimacy. These concepts often lack intuitive physical counterparts, and their meanings can shift in different contexts. Logical representations force researchers to clearly define the connotations and extensions of these abstract concepts and establish logical relationships between them, making theoretical discussions more precise and operational[29].

The first-order logic description of social structure is similar to that of economic structure. Here, we only give the theorem.

Theorem 11 (First-order representability of finite social structures)**.** *Let $\mathfrak{S} = (A, R_1, R_2, \ldots, F_1, F_2, \ldots, P_1, P_2, \ldots)$ be a finite social structure, that is:*

1. $|A| < \infty$ *(finite agent)*
2. *Each relation R_i and function F_j is defined over a finite domain and has corresponding predicate and function representations in the base language.*
3. *The process P_k involves finite states and finite time.*

Then there exists a set of first-order formulas Φ such that:

$$\forall \mathfrak{T}, \mathfrak{T} \models \Phi \iff \mathfrak{T} \cong \mathfrak{S} \tag{37}$$

Similarly, considering the limitations of social structure, we can draw a conclusion.

Theorem 12. *Within the theoretical framework of first-order and higher-order logic, almost all social structures can be fully characterized.*

Through logical representation, sociology can not only describe social phenomena more accurately, but also discover hidden social laws, providing more powerful theoretical tools for understanding and improving social structure.

Here, we give an example of logical representation.

Table 2. Semantic interpretation table of sociological predicates.

predicate	Semantic meaning
$Friend(x, y)$	x and y are friends
$Smokes(x)$	x smoking
$Influenced(x, y)$	x is affected by y
$HigherSmokingProbability(x)$	x has a higher probability of smoking
$SocialNetwork(x, y)$	x and y are in the same social network

Through the given predicate, we can express the state of the smoking phenomenon in sociology.Here, HigherSmokingProbability(x) is abbreviated as HSP(x).

Definition 21 (Expressions related to smoking)**.**

$$\phi_1 = \forall x, y(Friend(x, y) \to Friend(y, x)) \text{ (Friendship is symmetrical)} \tag{38}$$

$$\phi_2 = \forall x, y(Friend(x, y) \land Smokes(y) \to Influenced(x, y)) \text{ (Friends' smoking has an impact)} \tag{39}$$

$$\phi_3 = \forall x(Influenced(x, y) \to HSP(x)) \text{ (Increases the probability of smoking)} \tag{40}$$

$$\phi_4 = \forall x, y, z(Friend(x, y) \land Friend(y, z) \to SocialNetwork(x, z)) \text{ (Forming a social network)} \tag{41}$$

5. Computer Field State Expression

In computer science, logic plays an irreplaceable role as a fundamental tool for expressing states. In program verification, Hoare logic precisely describes the state of each execution point of a program through preconditions, postconditions, and invariants, enabling rigorous proof of program correctness. In database systems, first-order logic is not only used to define the semantics of query languages, but also characterizes the legal state space of data through integrity constraints. In the field of formal methods, temporal logic (such as LTL and CTL) can express the dynamic behavior and safety properties of systems in the time dimension, providing a mathematical foundation for modeling concurrent and real-time systems. Planning problems in artificial intelligence are essentially about finding a path from an initial state to a target state in the state space described by logic. In hardware design, Boolean logic directly corresponds to the physical state of circuits, enabling the design of complex digital systems. This abstract expressive power of logic not only provides a unified mathematical framework for modeling complex systems but, more importantly, enables automated reasoning and verification. From compiler optimization analysis to operating system resource management, from network protocol correctness verification to interpretability analysis of machine learning models, logic plays a critical role in translating intuitive concepts into computable forms[30–32].

5.1. Boolean Algebra and the Formalization of Computer Systems

Boolean algebra and logical representation are core areas of computer science and mathematical logic[33,34]. First, we verify that Boolean algebra can be formalized using first-order logic.

Theorem 13 (The expressive power of Boolean algebra and first-order logic)**.** *All axioms and operations of Boolean algebra can be expressed using first-order logic (FOL).*

Proof of Theorem13. Boolean algebra is defined as: a set B, binary operations $\land, \lor$, unary operations $\neg$, constants $0, 1$, and the following axioms:

$$\forall a, b \in B : \quad a \land b = b \land a \tag{42}$$

$$\forall a \in B : \quad a \lor \neg a = 1 \tag{43}$$

$$\forall a, b, c \in B : \quad a \land (b \lor c) = (a \land b) \lor (a \land c)\text{etc.} \tag{44}$$

$$\tag{45}$$

Given that Boolean algebra is a finite mathematical structure, all its structures and properties can be formalized using first-order logic. Therefore, the theorem is proved. □

Boolean algebra is the foundation of computer systems. The logic gates in digital circuits, conditional branching in programs, propositional calculus, and automata are all based on Boolean algebra. Operations, states, and transitions can all be reduced to Boolean algebraic expressions.

Because the entire content of Boolean algebra can be expressed using first-order logic, and computer systems can be reduced to Boolean algebraic structures, its core content can also be expressed using first-order logic. Practical applications such as model checking, hardware verification, and theorem proving have extensively employed first-order logic for modeling and reasoning.

Theorem 14. *Computer systems can be formally expressed in first-order logic.*

doi:10.20944/preprints202508.0939.v1

5.2. Predicate Logic Description of a Turing Machine (TM)

Definition 22. *Similar to finite state machines, Turing machines can be formally expressed. A deterministic Turing machine (DTM) can be represented as a seven-tuple:*

$$M = (Q, \Gamma, \Sigma, \delta, q_0, q_{accept}, q_{reject}) \tag{46}$$

Where:

- *Q is a finite state set.*
- *Γ is the set of tape symbols (including the blank symbol $\sqcup$).*
- *$\Sigma \subseteq \Gamma$ is the set of input symbols (excluding $\sqcup$).*
- *$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the state transition function, where L and R indicate whether the read/write head moves left or right.*
- *$q_0 \in Q$ is the initial state.*
- *$q_{accept}, q_{reject} \in Q$ are the accept and reject states, respectively.*

To represent a Turing machine, we can define the following predicates and give the corresponding state expressions:

Definition 23 (Predicate Definition and State Expression)**.** *The predicates and states in a Turing machine can be expressed as:*

- ***State**: $State(q)$ indicates that q is a state.*

$$\phi_1 = \forall q \; (State(q) \to q \in Q) \tag{47}$$

- ***Tape Symbol**: $Typesymbol(a)$ indicates that a is a tape symbol.*

$$\forall a \; (TypeSymbol(a) \to a \in \Gamma) \tag{48}$$

- ***Tape content**: $Cell(t, p, a)$ indicates that a is the tape symbol at time t and position p.*

$$\forall t \forall p \forall a \; (Cell(t, p, a) \to a \in \Gamma) \tag{49}$$

- ***Read/Write Head Position**: $Head(t, p)$ indicates that the head is at position p at time t.*

$$\forall t \exists p \; Head(t, p) \tag{50}$$

- ***Transition function**: $Transtion(q, a, q', a', d)$ represents state transition, where d is the direction.*

$$\forall q \forall a \forall q' \forall a' \forall d \; (Transition(q, a, q', a', d) \leftrightarrow \delta(q, a) = (q', a', d)) \tag{51}$$

- ***Initial state**: $Initial(q)$ represents the initial state q.*

$$\exists q_0 \; (Initial(q_0) \wedge State(q_0)) \tag{52}$$

- ***Accept state (similar to the rejection state)**: $Accept(q)$ indicates that q is an accepting state.*

$$\forall q \; (Accept(q) \leftrightarrow q = q_{accept}) \tag{53}$$

Based on the above definition of predicates and the corresponding Turing machine state representation and state transition representation, we derive the theorem:

Theorem 15 (Turing Machine State Representation)**.** *For any Turing machine, its input, output, and state transition behavior over a relevant time set can be described using a state set.*

Before Turing, concepts such as "computation," "algorithm," and "efficient process" were intuitive. Mathematicians knew what computation was, but could not give a rigorous mathematical definition. The logical representation of the Turing machine was the first to transform these intuitive concepts into precise mathematical objects: state sets, symbol sets, transition functions, initial states, and accepting states. This formalization enables us to use mathematical methods to study the properties of computation itself[35].

5.3. Mathematical Formalization of Neural Networks

Neural networks (NNs) are machine learning models that mimic the structure and function of biological neural systems, capable of learning complex patterns from data and making predictions or decisions[36]. They are a core technology in deep learning and are widely used in fields such as computer vision, natural language processing, and speech recognition.

First, let's briefly introduce neural networks. A neural network can be represented as a function $\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$, whose hierarchical structure is decomposed into:

$$\mathcal{N}(x) = \sigma_L(W_L \cdot \sigma_{L-1}(W_{L-1} \cdots \sigma_1(W_1 x + b_1) \cdots) + b_L) \tag{54}$$

Where: input layer: $x \in \mathbb{R}^n$, hidden layer: $h_l = \sigma_l(W_l h_{l-1})$, output layer: $\mathcal{N}(x) \in \mathbb{R}^m$, weight matrix: $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$, bias vector: $b_l \in \mathbb{R}^{d_l}$, activation function: $\sigma_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$

Single neuron calculation:

$$\sigma(w^T x + b) = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{55}$$

Neural networks, by simulating the connections of the human brain, enable efficient modeling of complex data. With advances in computing power and algorithms (e.g., GPUs and attention mechanisms), their capabilities are continuously expanding, becoming the driving force behind the AI revolution.

Next, we demonstrate that the relevant aspects of neural networks can be formally expressed.

Proposition 1. *Neural networks can be formally expressed in first-order and higher-order logic.*

Proof of Proposition1. The first step is to logically represent a single neuron in a neural network.

Definition 24 (Neuron Triplet)**.** *A neuron can be represented as $\mathcal{N} = (w, b, \sigma)$, where $w \in \mathbb{R}^n$ represents the weight vector, $b \in \mathbb{R}$ represents the bias term, and $\sigma : \mathbb{R} \to \mathbb{R}$ represents the activation function.*

Using the neuron triple model, we can see that the neuron input-output relationship can be logically represented as follows:

$$\forall x \in \mathbb{R}^n, \exists z, a \in \mathbb{R}, \left(z = \sum_{i=1}^{n} w_i x_i + b\right) \wedge (a = \sigma(z)) \tag{56}$$

Next, we express the activation function in the neural network. Because activation functions possess strong mathematical properties and are finite mathematical structures, they are naturally amenable to logical expression.

Finally, we demonstrate the logical representation of the network topology. Taking a simple fully connected layer as an example:

- **Feedforward Network**: $Connected(u, v)$ indicates the connection between u, v, and $Layer_l$ indicates the lth layer.

$$\begin{gathered} \forall l \in \{1, ..., L\}, \forall u \in \text{Layer}_l, \forall v \in \text{Layer}_{l+1}, \\ \text{Connected}(u, v) \wedge \neg\exists k < l, \text{Connected}(v, \text{Layer}_k). \end{gathered} \tag{57}$$

- **Hierarchical Combination**: $Network(x)$ indicates the hierarchical combination structure of x.

$$\exists \text{Network} : (\mathbb{R}^{d_0} \to \mathbb{R}^{d_L}), \ \text{Network}(x) = \sigma_L \circ W_L \circ \cdots \circ \sigma_1 \circ W_1(x) \tag{58}$$

- **Combinatorial Completeness**: If the lth layer can be represented as Φ_l, then the $l+1$th layer can be represented as:

$$\Phi_{l+1} = \exists y_l, \Phi_l(x, y_l) \wedge (y_{l+1} = \sigma_{l+1}(W_{l+1}y_l + b_{l+1})) \tag{59}$$

Thus, we have proved the conclusion. □

According to the theorem, from a theoretical perspective, there is a profound equivalence between neural networks and logical systems. Neural networks can discover strategic patterns that humans have never discovered. If these patterns can be expressed in logical form, they may be transformed into verifiable scientific theories. This fusion is not just a technological advancement; it also represents a deepening of our understanding of the nature of intelligence: intelligence requires both the ability to learn from experience and the ability to reason based on rules, and the logical representation of neural networks is the key bridge connecting the two.

5.4. Formal Expression of States in Computer Science

According to previous proofs, computer systems, Turing machines, and neural networks can all be formalized using first-order and higher-order logic. Scholars have also formalized phenomena such as computational concepts, programming languages, and algorithmic processes.[11,37,38]. Furthermore, emerging problems in computer science can be gradually formalized using the proof process of neural networks. Based on this, we draw a comprehensive conclusion.

Theorem 16. *All phenomena in computer science can be formalized using first-order and higher-order logic.*

Theorem 16 demonstrates that logic provides a unified framework for expressing different computational models. Logical formalization has transformed computer science from an engineering craft into a rigorous scientific discipline, providing powerful tools for understanding and controlling complex systems.

6. State Expression in Natural Language Domain

Natural languages, such as Chinese, English, and Arabic, are the languages humans use in everyday life. They are essential tools for communicating ideas and conveying information. In contrast to formal languages, they possess unique properties. Natural languages are richly expressive, capable of expressing the myriad worlds, complex emotions, and abstract concepts. This chapter introduces Montague semantics, explores the rules governing the grammar and semantic translation of natural languages, and achieves a formal understanding of natural languages through the principles of formalized mathematical logic.

6.1. Montague Semantics

Montagu semantics, also known as Montagu grammar, is a formalized approach to the study of natural semiotics, particularly intensional semantics. It represents a new stage in the development of linguistics and logic.

Montagu's research began with the concept of categories, dividing English syntax into distinct categories. He then established 17 general rules for the formation of all English syntax. He then defined meaningful expressions, used recursion to represent meaningful sets, and explained them using model theory. Finally, Montagu defined 17 corresponding rules for semantic translation, corresponding to the 17 rules for the formation of syntax[39,40].

Based on Montagu's work, Bennett conducted further research. He refined the English categories, dividing adjectives into separate categories; introduced more complex grammatical and semantic translation rules, expanding the rules from 17 to 35, and provided a solid theoretical foundation for language research using Montagu semantics[41].

6.2. Study of English Ambiguity

Correctly handling linguistic ambiguity is an important indicator of semantic comprehension. Here, we use Montague Grammar to provide an interpretation of ambiguity.

"At least one person likes the book" is a common ambiguous phrase in English. The ambiguity of "at least one person likes the book" stems from the ambiguity of the quantifier scope. Without context, "the book" could refer to a specific book or to a general term.

The semantics of the sentence **"At least one person likes the book"** can be formally modeled using the λ calculus, with two possible interpretations: a broad interpretation and a narrow interpretation.

In the broad interpretation, "the book" refers to a specific book b. The logical form of the entire sentence is:

$$\exists x[man'(x) \wedge like'(\hat{}b')(x)] \tag{60}$$

Where:

- $man'(x)$ means x is a person;
- $like'(\hat{}b')(x)$ means x likes a specific book b.

The specific λ calculus combination process is as follows:

$$\begin{aligned}
\text{Like} &: like' \\
\text{The book} &: b' \\
\text{Like the book} &: like'(\hat{}b') \\
\text{At least one person} &: \lambda P.\exists x[man'(x) \wedge P\{x\}] \\
\text{Combination result} &: \lambda P.\exists x[man'(x) \wedge P\{x\}](\hat{}like'(\hat{}b')) \\
\lambda\ \text{transposition} &: \exists x[man'(x) \wedge \hat{}like'(\hat{}b')\{x\}] \\
\text{Bracket convention} &: \exists x[man'(x) \wedge \check{}\hat{}like'(\hat{}b')(x)] \\
\text{Top and bottom elimination} &: \exists x[man'(x) \wedge like'(\hat{}b')(x)]
\end{aligned}$$

In the narrow-scope interpretation, "the book" is not a specific object but a quantifiable range. The logical form of the sentence is:

$$\exists x[man'(x) \wedge (like'(x, \hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))] \tag{61}$$

Where:

- $book'(b)$ indicates that b is a book;
- $(like'(x, \hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))$ indicates that there exists at least one person x, there exists a book b, and this person likes book b.

The specific λ calculus combination process is as follows:

$$\begin{aligned}
\text{Like} &: like' \\
\text{At least one person} &: \lambda P.\exists x[man'(x) \wedge P\{x\}] \\
\text{The book exists} &: \lambda Q.\exists b[book'(b) \wedge Q\{b\}] \\
\text{Like the book} &: like'(\hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]) \\
\text{Combination result} &: \lambda P.\exists x[man'(x) \wedge P\{x\}](\hat{}like'(\hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}])) \\
\lambda\text{Transposition} &: \exists x[man'(x) \wedge (\hat{}like'(\hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))\{x\}] \\
\text{Bracket convention} &: \exists x[man'(x) \wedge (\check{}\hat{}like'(\hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))(x)] \\
\text{Up and Down Elimination} &: \exists x[man'(x) \wedge (like'(\hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))(x)] \\
\text{Relational notation} &: \exists x[man'(x) \wedge (like'(x, \hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))]
\end{aligned}$$

By modeling the λ calculus, the sentence "At least one person likes this book" can capture the following two semantic interpretations:

1. **Broad interpretation**: There is at least one person who likes this particular book:

$$\exists x[man'(x) \wedge like'(\hat{}b')(x)] \tag{62}$$

2. **Narrow interpretation**: There is a book and at least one person likes it:

$$\exists x[man'(x) \wedge (like'(x, \hat{}\lambda Q.\exists b[book'(b) \wedge Q\{b\}]))] \tag{63}$$

Similarly, the ambiguous phenomenon of "Every student read a book" has been extensively studied and interpreted. The quantifier scope ambiguity involved in this issue is an active frontier in semantic research, continuing to drive theoretical and technological developments[42]. Studying such issues can further advance the research and development of natural semantics.

6.3. Optimizing Syntax and Semantic Translation Rules

Although Montagu, Bennett, and others have established detailed rules for English semantics and grammar, which later generations can simply apply directly, many issues still need to be resolved in the actual translation process.

6.3.1. Conjunction Rules

Among Montagu's 17 rules, S11 and S12 deal with conjunction, defining parallel sentences and parallel verbs, respectively. Among Bennett's 35 rules, S28 deals with conjunction, defining only parallel verbs. However, as we know, conjunctions of nouns and noun phrases occur very frequently in natural language. Surprisingly, neither Montagu nor Bennett provide corresponding grammatical rules for this phenomenon. This is because conjunctions of nouns and noun phrases require the verb to become plural, and to simplify expression, no corresponding grammatical rules were defined. However, given the high frequency and widespread use of nouns and noun phrases, and to help beginners better grasp the relevant content, we provide the following additional rules:

Definition 25 (Grammatical Rules for Conjunction of Nouns and Noun Phrases)**.** *If $\alpha, \beta \in P_{CN}$, then $F(\alpha, \beta) \in P_{CN}$. If $\alpha, \beta \in P_T$, then $F(\alpha, \beta) \in P_T$. Here, $F(\alpha, \beta) = \alpha$ and β. And when the object of the conjunction becomes the subject, the corresponding verb becomes plural.*

Definition 26 (Translation Rules for Conjunctions of Nouns and Noun Phrases)**.** *If $\alpha, \beta \in P_{CN}$ or $\alpha, \beta \in P_T$, and α, β is translated as α', β', then α and β is translated as $\lambda P[\alpha'(P) \wedge \beta'(P)]$. When the conjunction object serves as the subject, the corresponding verb is translated into its plural form.*

6.3.2. Adjective Rules

Among Bennett's 35 rules, the one concerning adjectives is S10. We give its original definition:

Definition 27 (S10)**.** *If $\gamma \in P_{AJ}$ and $\zeta \in P_{CN}$, then $F_9(\gamma, \zeta) \in P_{CN}$, where*

(a) *if γ contains an occurrence of a member of $B_{AJ/T}$, then $F_\varphi(\gamma, \zeta) = \zeta\gamma$;*

(b) *otherwise $F_9(\gamma, \zeta) = \gamma\zeta$.*

Bennett argues that using S10 can resolve the English grammatical phenomenon of adjective + noun. However, let's consider the following example: John's mother. According to Bennett's definition, John's does not fall into the basic category of adjectives, so S10 cannot be used for translation. We must instead use S5.

Definition 28 (S5)**.** ***S5.*** *If $\zeta \in P_{CN/T}$ and $\alpha \in P_T$, then $F_5(\zeta, \alpha) \in P_{CN}$, where*

(a) *if $\alpha = he_n$, then $F_5(\zeta, \alpha) = \zeta * him$;*

(b) *otherwise $F_5(\zeta, \alpha) = \zeta\alpha$.*

Consider John's mother to be equivalent to mother of John. This can be translated:

$$\begin{aligned} \text{mother} &: \textit{mother}' \\ \text{John} &: \lambda P[P\{j\}] \\ \text{mother of John} &: \textit{mother}'(\hat{}(\lambda P[P\{j\}])) \end{aligned}$$

While there's certainly nothing wrong with translating "John's mother" this way, treating "John's mother" and "mother of John" as equivalent loses the distinction between the two grammatical structures. Furthermore, the resulting translation is often less concise and clear, hindering the reader's intuitive understanding. Therefore, we provide supplementary rules for these situations.

Definition 29 (Grammar rule)**.** *If $\alpha \in P_{CN}$ or $\alpha \in P_T$, then $G_2(\alpha) \in P_{AJ}$, where*

(a) *If $\alpha = he_n$, then $G_2(\alpha) = his_n$;*

(b) *Otherwise $G_2(\alpha) = \alpha's$.*

Definition 30 (Semantic Translation Rules)**.** *If $\alpha \in P_{CN}$ or $\alpha \in P_T$, and α is translated as α'', then*

(a) *his_n is translated as $his_n{}'$;*

(b) *John's is translated as $\lambda P[P\{j's\}]$.*

(b) *In other cases, $\alpha's$ is translated as $\alpha''s'$.*

Using the new rules, we can retranslate John's mother:

$$\begin{aligned} \text{mother} &: \textit{mother}' \\ \text{John's} &: \lambda P[P\{j's\}] \\ \text{John's mother} &: \lambda P[P\{j\}](\hat{}\textit{mother}') \\ \lambda \text{ transposition} &: \hat{}\textit{mother}'\{j\} \\ \text{Brackets, upper and lower rules} &: \textit{mother}'(j's) \end{aligned}$$

In comparison, the translation using the new rules is more concise and clear, making it easier for readers to grasp the grammatical structure.

6.3.3. Clause Rules

For the sake of brevity, neither Montagu nor Bennett discussed clause rules in detail, instead focusing on the typical relation “such that.” Some might question whether each clause has a specific function and introductory phrase, expressing different logical relationships depending on the context. Since their functions are not identical to “such that,” does this mean that the same rules cannot be applied universally?

Indeed, “such that” primarily expresses a result or condition. Commonly used attributive clauses, such as modifiers and adverbial clauses, often express time or cause, differ significantly from “such that.” However, these commonly used clauses are structurally simpler than standard relative clauses. Generally speaking, one can emulate Montagu’s approach to translation by modifying or simplifying Montagu’s rules or by transforming the clause form.

Here, we provide a case study: the man whom Mary loves.

man : man'

the man : $\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}]$

Mary : $\lambda Q[Q\{m\}]$

loves : $love'$

loves the man : $love'({}^{\wedge}(\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}]))$

Mary loves the man : $\lambda Q[Q\{m\}]({}^{\wedge}love'({}^{\wedge}(\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}])))$

λTransposition : ${}^{\wedge}\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}]))\{m\}$

Brackets, upper and lower rules : $\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}]))(m)$

λTransposition : $\exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge m\{y\}]))$

Combined results:

$$\begin{aligned} &\lambda x_n[(\lambda P \exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge P\{y\}])(x_n) \wedge \\ &(\exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge m\{y\}]))] \end{aligned} \tag{64}$$

After λ transposition, brackets, and upper and lower rules, the final result is:

$$\begin{aligned} &\lambda x_n[\exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge {}^{\vee}x_n(y)]) \wedge \\ &(\exists y[\forall x[man'(x) \leftrightarrow x = y] \wedge {}^{\vee}m(y)]))] \end{aligned} \tag{65}$$

The core insight of Montague semantics lies in placing natural language within a formal framework as rigorous as that of mathematics and logic. By supplementing these rules, scholars can better understand natural language through logic. Formal logic is not just an abstract mathematical tool; it is a key approach to understanding and modeling the most complex human cognitive abilities. As artificial intelligence progresses toward general intelligence, the formal methodology represented by Montague semantics will continue to play an irreplaceable role.

6.4. Formalization of Natural Languages

Scholars such as Montague have provided comprehensive tools and detailed introductions to the formalization of English, a natural language. Furthermore, combined with the rules subsequently added by scholars, it can be argued that English can be fully formalized. Similarly, other natural languages can be formalized using similar methods. Alternatively, given the intertranslatability between English and other languages, they can be directly converted into English for formalization.

In short, we can conclude.

Theorem 17. *All natural languages can be fully formalized using first-order and higher-order logic with appropriate expansion of symbols and rules.*

The importance of formalization of natural languages ultimately lies in the scientific method it provides for understanding the essence of human intelligence. Language is not only a tool for communication but also a vehicle for thought, a container for knowledge, and a medium for the transmission of culture. Formalizing language is a mathematical modeling of human cognitive abilities and a scientific exploration of the essence of intelligence.

7. Conclusion and Outlook

This paper systematically investigates the logical formalization of object states, proposing and demonstrating a rigorous and universally applicable formalization framework for revealing the nature of information and its interdisciplinary applications. The paper first reviews classical information theory and its shortcomings, noting the current lack of a unified and rigorous mathematical definition of the core concept of "state."

To this end, this paper establishes a universal representation system for information states based on first-order and higher-order predicate logic, combined with modal logic and calculus, addressing the current lack of formal representations for states. This paper enumerates typical states from various fields, including mathematics, economics, sociology, computer science, and natural language, and rigorously proves that these states can be formalized using first-order and higher-order logic. Because the proof methods used in these fields can be extended to other fields, this paper demonstrates a universally applicable method for representing states in any domain: formalization using first-order and higher-order predicate logic.

Thus, first-order and higher-order logic are not merely technical tools but also cognitive bridges for understanding the unity of the world. They integrate scattered fragments of knowledge into coherent theoretical systems, elevate local understandings of phenomena into global conceptual frameworks, and transform static descriptions of states into dynamic reasoning processes. In this sense, logic has truly become a universal bridge connecting states across various fields, a universal language that enables communication across fields and provides humanity with the most fundamental and powerful mathematical tools for understanding and transforming the world.

Through the formalization of states, objective information theory has been further refined and developed, deepening research on the nature of information and expanding its scope from the classic Shannon framework. Many pressing problems in information science can be transformed into logical problems. By studying the properties of logical language and drawing on proven conclusions and axioms from logic, we can clarify the meaning of information and guide the development of information research. **Funding:** This research received no external funding

Acknowledgments: I would like to thank Professor Xu for his guidance all the time. He has provided me with great help in writing, content organization, topic selection, etc. Without him, I would not have been able to complete this paper.

Conflicts of Interest: The authors declare no conflicts of interest. The authors have identified and declared that there are no personal circumstances or interests that may be perceived asinappropriately influencing the representation or interpretation of the reported research results. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

OIT	Objective Information Theory
FOL	First-order predicate logic
HOL	Higher-order predicate logic

References

1. N. WIENER. *Cybernetics or Control and Communication in the Animal and the Machine*. The MIT Press, Cambridge, 2019.
2. C. E. Shannon. The mathematical theory of communication. *Bell Syst Tech*, 27:379–423, 1948.
3. J. von NEUMANN. *Mathematische Grundlagen der Quantenmechanik*, volume 38. Springer Berlin Heidelberg, Berlin, Heidelberg, 1971.
4. A. N. KOLMOGOROV. Three approaches to the quantitative definition of information. *International journal of computer mathematics*, 2(1-4):157–168, 1968.
5. MCGOWAN A T. User and information dynamics: Managing change. *Bulletin of the Medical Library Association*, 78:327–329, 1990.
6. XU. J, MA. X, SHEN. Y, et al. Objective information theory: A sextuple model and 9 kinds of metrics. *2014 Science and information conference*. IEEE, pages 793–802, 2014.
7. XU. J, MA. X, TANG. J, et al. Research on model and measurement of objective information. *Science China Information Sciences*, 45(3):336–353 (In Chinese), 2015.
8. XU. J, LIU. Z, WANG. S, et al. Foundations and applications of information systems dynamics. *Engineering(Beijing, China)*, 27:254–265, 2023.
9. Alan G Hamilton. *Logic for mathematicians*. Cambridge University Press, 1988.
10. P. B. ANDREWS. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof: vol 27*. Dordrecht: Springer Netherlands, 2002.
11. Jianfeng Xu. Information science principles of machine learning: A causal chain meta-framework based on formalized information mapping. *arXiv preprint arXiv:2505.13182*, 2025.
12. Richard G Swan. *K-theory of finite groups and orders*, volume 149. Springer, 2006.
13. Rudolf Lidl and Harald Niederreiter. *Finite fields*. Number 20. Cambridge university press, 1997.
14. Leonid Libkin. *Elements of finite model theory*, volume 41. Springer, 2004.
15. Chen Chung Chang and H Jerome Keisler. *Model theory*, volume 73. Elsevier, 1990.
16. Richard Dedekind. Was sind und was sollen die zahlen? In *Was sind und was sollen die Zahlen?. Stetigkeit und Irrationale Zahlen*, pages 1–47. Springer, 1965.
17. Giuseppe Peano. *Arithmetices principia: Nova methodo exposita*. Fratres Bocca, 1889.
18. Dana Scott. Logic with denumerably long formulas and finite strings of quantifiers. In *The theory of models*, pages 329–341. Elsevier, 2014.
19. Gerard Debreu. *Theory of value: An axiomatic analysis of economic equilibrium*, volume 17. Yale University Press, 1959.
20. Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
21. John Geanakoplos. Three brief proofs of arrow's impossibility theorem. *Economic Theory*, 26(1):211–215, 2005.
22. Ulle Endriss. Logic and social choice theory. 2012.
23. W Brian Arthur, Steven N Durlauf, and David A Lane. The economy as an evolving complex system ii. adison wesley. *Reading, MA*, 1997.
24. Jaakko Hintikka and Jack Kulas. *Anaphora and Definite Descriptions: Two Applications of Game-Theoretical Semantics*, volume 26. Springer Science & Business Media, 1985.
25. Daniel M Hausman. *The inexact and separate science of economics*. Cambridge University Press, 2023.
26. James Samuel Coleman. Introduction to mathematical sociology. 1964.
27. Stanley Wasserman and Katherine Faust. Social network analysis: Methods and applications. 1994.
28. Patricia H Thornton, William Ocasio, and Michael Lounsbury. *The institutional logics perspective: A new approach to culture, structure, and process*. Oxford University Press, 2012.
29. Stephen P Borgatti, Martin G Everett, Jeffrey C Johnson, and Filip Agneessens. *Analyzing social networks using R*. Sage, 2022.

30. Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press, 2004.
31. Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.
32. Benjamin C Pierce. *Types and programming languages*. MIT press, 2002.
33. George Boole. *The mathematical analysis of logic*. CreateSpace Independent Publishing Platform, 1847.
34. George Boole. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*, volume 2. Walton and Maberly, 1854.
35. Alan Mathison Turing et al. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.
36. Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
37. Glynn Winskel. *The formal semantics of programming languages: an introduction*. MIT press, 1993.
38. Hartley Rogers Jr. *Theory of recursive functions and effective computability*. MIT press, 1987.
39. Richard Montague. English as a formal language. 1970.
40. Richard Montague. The proper treatment of quantification in ordinary english. In *Approaches to natural language: Proceedings of the 1970 Stanford workshop on grammar and semantics*, pages 221–242. Springer, 1973.
41. Michael Bennett. A variation and extension of a montague fragment of english. In *Montague Grammar*, pages 119–163. Elsevier, 1976.
42. Robin Cooper. *Quantification and syntactic theory*, volume 21. Springer Science & Business Media, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.