Towards the Metaontology Implementation	
Version	1.1
Date	13.06.2025
Author	Cognito One SAS

Abstract

This document briefly describes the problem of determinism's limitations, the possibility of overcoming them with deterministic devices, the formulation of related tasks, and associated epistemological questions.

As a solution, this paper proposes the formalization of a (meta)ontology and an algebra of monads. These are formalisms that allow for operating on entities and methods at the same level of abstraction as data.

Within this (meta)ontological framework, we propose definitions for a priori and a posteriori data, criteria for truth and verification, the boundaries of rational cognition, and the fundamental limits of devices that lack consciousness.

We also propose a formalization of a priori operations that allows deterministic devices a partial and limited means of bypassing the formulated constraints of determinism.

1. Problem and Task Formulation

1.1 Limitations of Determinism

Before formulating the core problem and its associated tasks, we offer a brief introduction to substantiate the limitations of current paradigms. From these considerations, the problem emerges, along with the need for new solutions.

Reversibility and Informational Isentropy

Lemmas 1-3 are not proven but are largely formulated directly from the definitions of informational entropy, surjection, injection, and bijection.

Let us consider discrete, non-empty sets of values $P,\ Q$ with defined sets of positive probabilities $p,\ q$ and a computation $f\colon\ P\ \to Q$

Lemma 1. Without surjection, the informational entropy of an injective computation increases.

Assume the condition of injection is met for all $n\ =\ |P|$ input values. For convenience, let the value with no preimage be denoted as Q_{n+1}. Then:

$$f\colon \left\{P_1,\ P_2,\ ...,\ P_n\right\}\ \to\ \left\{Q_1,\ Q_2,\ ...,\ Q_n,\ Q_{n+1}\right\}$$

$$\forall i\ \le\ n\colon\ p_i > q_i \qquad q_{n+1} = 1\ -\ \sum_{i=1}^{n} q_i$$

Due to the appearance of a new possible value, the probabilities qi have decreased. From Shannon's definition of informational entropy:

$$H(Q)\ -\ H(P)\ = \sum_i p_i\, log\, p_i - \sum_j q_j\, log\, q_j\ =\ log\frac{\prod_i p_i^{\,p_i}}{\prod_j q_j^{\,q_j}}\ =$$

$$=\ log \prod_{i=1}^{n} \left(\frac{p_i}{q_i}\right)^{p_i - q_i} \frac{1}{q_{n+1}^{\,q_{n+1}}}\ >\ log1\ =\ 0$$

With *k* output values lacking preimages, uncertainty only increases:

$$H(Q)\ -\ H(P)\ =\ log \prod_{i=1}^{n} \left(\frac{p_i}{q_i}\right)^{p_i - q_i} \frac{1}{q_{n+1}^{\,q_{n+1}} \times q_{n+2}^{\,q_{n+2}} \times ... \times q_{n+k}^{\,q_{n+k}}}\ >\ log1\ =\ 0$$

In informational terms, when the range of values increases relative to the domain, more information is required to encode the result than the argument. In statistical terms, when the set of a system's possible states increases, the function's range contains more uncertainty than its initial domain.

Lemma 1 can also be interpreted as a condition of determinism: a computation should not return "extraneous" or "independent" results from its arguments. An argument here could be the current time, a pseudo-random sequence generator key, neural network weights, etc.

If the number of possible outcomes exceeds the number of initial states (due to spontaneity, noise, errors, etc.), it is impossible to uniquely or deterministically associate the same initial state with the same outcome.

Lemma 2. **Without injection, the informational entropy of a surjective computation decreases.**

Suppose the injection condition fails for only one input value. Let's denote it as P_{n+1}, where $n = |P| - 1$:

$$f: \{P_1, P_2, ..., P_n, P_{n+1}\} \rightarrow \{Q_1, Q_2, ..., Q_n\}$$
$$\forall i < n: \; q_i = p_i \;\; ; \;\; q_n = p_n + p_{n+1}$$

From Shannon's definition of informational entropy:

$$H(Q) - H(P) = \sum_i p_i \log p_i - \sum_j q_j \log q_j = \log \frac{\prod_i p_i^{p_i}}{\prod_j q_j^{q_j}} =$$

$$= \log \frac{p_n^{p_n} \times p_{n+1}^{p_{n+1}}}{(p_n + p_{n+1})^{(p_n + p_{n+1})}} = \log \left[\left(\frac{p_n}{p_n + p_{n+1}} \right)^{p_n} \left(\frac{p_{n+1}}{p_n + p_{n+1}} \right)^{p_{n+1}} \right] < \log 1 = 0$$

If the condition fails for *k* input values, the uncertainty only decreases:

$$f: \{P_1, P_2, ..., P_n, P_{n+1}, ... , P_{n+k}\} \rightarrow \{Q_1, Q_2, ..., Q_n\}$$
$$\forall i < n: \; q_i = p_i \;\; ; \;\; q_n = \sum_{j=n}^{n+k} p_j$$
$$H(Q) - H(P) = \log \frac{p_n^{p_n} \times p_{n+1}^{p_{n+1}} \times .. \times p_{n+k}^{p_{n+k}}}{(p_n + p_{n+1} + p_{n+k})^{(p_n + p_{n+1} + p_{n+k})}} =$$

$$= log\left[\left(\frac{p_n}{p_n + p_{n+1} + ... + p_{n+k}}\right)^{p_n}\left(\frac{p_{n+1}}{p_n + p_{n+1} + ... + p_{n+k}}\right)^{p_{n+1}} ... \left(\frac{p_{n+k}}{p_n + p_{n+1} + ... + p_{n+k}}\right)^{p_{n+k}}\right] < 0$$

The same holds true if the condition fails for *m* groups of different input values mapping to *m* corresponding values from Q.

In informational terms, when the range of values decreases relative to the domain, less information is required to encode the result than the arguments. In statistical terms, when the set of a system's possible states decreases, the function's range contains less uncertainty than its initial domain.

Lemma 3. **A computation is informationally isentropic if and only if it is reversible.**

Analyzing the change in entropy when both surjection and injection are absent is non-trivial. It requires considering how the uncertainty generated by the lack of surjection relates to the uncertainty eliminated by the lack of injection, a relationship that depends on the specific probability sets in each case.

Proving this lemma's sufficiency leads to a more fundamental question: do two different probability sets with identical entropy exist? Or two different systems with the same number of possible configurations?

$$\forall P \ \exists \ Q \neq P: \ H(P) = H(Q) \Leftrightarrow \prod_i p_i^{p_i} = \prod_j q_j^{q_j} \ , \ \sum_i p_i = \sum_j q_j = 1 \ ?$$

This complexity points to an ambiguity in the problem statement. The absence of surjection (non-determinism) implies that "noise" or "extraneous" results are independent of the input data. However, the condition of compensating for this uncertainty—balancing the eliminated and introduced uncertainty—establishes a dependence of the *entire* set of values (including "noise") on the arguments, which in turn implies surjection.

Therefore, a non-deterministic mapping is generally not isentropic, as the condition of isentropy itself imposes determinism. If we exclude non-deterministic mappings, it follows from Lemmas 1 and 2 that uncertainty remains constant if and only if the mapping is a bijection:

$$f: \{P_1, P_2, ..., P_n\} \rightarrow \{Q_1, Q_2, ..., Q_n\}$$
$$\forall i: \ q_i = p_i$$
$$H(Q) - H(P) = log1 = 0$$

In informational terms, a reversible mapping requires an equal amount of information to *encode the argument and the result. In statistical terms, a reversible function preserves the set of possible states and, consequently, the system's uncertainty.*

More broadly, the existence of an inverse function f^{-1} makes it impossible for f to generate or delete information. The definiteness of f^{-1} implies no uncertainty was eliminated in f, and vice versa, due to the bijective nature of the mapping. By analogy with thermodynamics, only a reversible process can be isentropic.

As an example, consider the addition and increment functions:
$add(x, y) = x + y$; add: $N \times N \rightarrow N \quad (N / \{1\})$
$incr(x) = x + 1$; $incr$: $N \rightarrow N \quad (N / \{1\})$

One could propose a mathematical construct to demonstrate the theoretical isentropy of addition. By defining a bijection between the set of ordered pairs of natural numbers and the set of their ordinal numbers, we can show that add: $N \rightarrow N$*. This defines addition as a bijection between the ordinal number of the argument pair and the result.*

Such a construct relies on the infinite, unbounded nature of an abstract set. In real computers, discrete cells and finite memory represent numbers. The information required to encode two equally sized integers cannot be losslessly compressed into one.

In a discrete domain, addition reduces uncertainty from two numbers to one. It is therefore informationally irreversible in the general case; one cannot deterministically recover the original pair of numbers from their sum alone, due to the difference in information content (i.e., the uncertainty of memory cell states).

The increment operation, a special case of addition with a fixed parameter, is informationally isentropic. Knowing the result makes it easy to calculate the argument. Unlike addition, it neither creates nor destroys information at a fundamental level. Both operations are widely used in practice.

Irreversibility of the Elimination of Uncertainty

Theorem on the irreversibility of the elimination of uncertainty

Even theoretical mathematics views functions as superpositions of arithmetic operations. On a real machine, any deterministic mapping $f\colon P \to Q$, $H(Q) < H(P)$, where P, Q are the sets of states of the machine's memory cells with input and output data, is not executable instantaneously. It is the result of a discrete sequence of state mappings as the machine executes each operation.

Besides the direct input-to-output mapping, any computational device has both a program (algorithm, environment description, etc.) and a computational context—an internal state needed to process data (registers, pointers, a call stack, local variables, etc.). Let's represent the set of all possible machine states at each operation as S_i.

Any abstract deterministic mapping $f\colon P \to Q$, $H(Q) < H(P)$ is physically computed on a machine as a sequence of deterministic mappings $g_i\colon S_i \to S_{i+1}$, where $H(S_{i+1}) \le H(S_i)$. These operations gradually eliminate the machine's initial uncertainty until a final result is reached.

The implementation of such a mapping f can be specified algorithmically, or as an environment description with an interpreter, or parametrically, as neural network weights. The essence of computation is the deterministic reduction of a larger set of input states to a smaller set of output states.

Because of the program and its context, the machine's set of possible internal states is larger than the set of states of the memory cells holding the result. That is, the machine's initial state has greater uncertainty than the input data: $H(S_0) > H(P)$. After the computation, the context is irreversibly lost, and only the output data Q remains.

Consequently, if the mapping itself reduces uncertainty by $H_c = H(P) - H(Q)$, the total uncertainty eliminated by the machine's work is:

$$H_t = \sum_i [H(S_i) - H(S_{i+1})] = H(S_0) - H(Q) > H_c = H(P) - H(Q)$$

Thus, even when computing a reversible mapping $f\colon P \to Q,\ where\ H_c = 0$, a deterministic device maps its internal state through a sequence of operations g_i such that $H_t > 0$.

Therefore, the implementation of a computation on a machine cannot be informationally isoentropic. The machine eliminates additional uncertainty from its own internal state, rendering the process irreversible (per Lemma 3).

A device computes any mapping through a sequence of operations, each involving an irreversible loss of information about the device's own state, making the entire computational process irreversible.

Even if a mapping is designed to output its internal state (e.g., via logs), this output must be accounted for in the final state set Q. Furthermore, it can never capture the complete internal state. For instance, when an output operation executes, the context of that very operation is itself irreversibly lost.

Note: the irreversibility of processes realization in time

Existence is discrete (quantized), so for its inert states at any moment i, we can define a discrete set of states P_i resulting from possible acts of change. These changes realize or implement time's potential.

Time's realization is perceived as a sequence of changes to discrete inertias, represented as mappings $f_i\colon P_i \to \{r_i\}$. Here, P_i is the set of all possible "future" states still outside of existence (in non-existence), and r_i is the single, actualized state of existence. This process involves an irreversible elimination of uncertainty $H(P_i)$, meaning the realization of existence is deterministic (Lemma 1).

The reverse mapping—from the single current state of existence to its set of possible "future" states—is non-deterministic due to injection, which increases uncertainty (Lemma 2):

$$g_i\colon \{r_i\} \to P_{i+1}$$

Time's potential is realized or implemented through acts of change that irreversibly discard information about other possible, but unrealized, changes. Determinism applies to discrete phenomena (material existence, capacities, inertias) because realizing their potential eliminates uncertainty.

However, with each act of time's realization, non-deterministic (i.e., undifferentiated, continuous) phenomena (potentials, flows and changes) generate a new set of possible future states from the current one. A predictive model's accuracy depends on how completely it defines potential states and the rules governing their changes. But as time's potential grows, the set of possible changes expands non-deterministically from non-existence. Thus, any static model inevitably loses accuracy in existence.

In other words, non-existence is non-deterministic and undifferentiated in its potentiation, while existence is deterministic and discrete in its realization.

Non-determinism in the Growth of Uncertainty Elimination

We will omit discussion of eliminating uncertainty in the device's physical state (design and assembly, power, cooling, etc.), as it requires linking informational and thermodynamic entropy, which is a highly complex and controversial topic.

The eliminated uncertainty of the machine's state, H_t, refers only to the non-physical, informational aspect of its operation. This includes the uncertainty of input data, context, instructions, and the machine's behavior, which was itself fixed by the act of programming (defining algorithms, training models, etc.).

Let's define the **coefficient of "useful" ordering** as the ratio of the informational uncertainty eliminated by the desired computation to the total informational uncertainty of the machine's state that had to be eliminated to perform that computation.

$$\eta_c = \frac{H_c}{H_t} < 1$$

The term "useful" is in quotes because its physical analogue is intentional (defined by an experimenter or client) and thus metaphysical. Is a forgotten heating iron "useful"? Are simulations of quantum computing on classical machines "useful"? To use such terms rigorously requires a strict demarcation between physics and metaphysics, which we avoid here by using the term conventionally.

The theorem on irreversibility implies that even in the informational aspect alone, the coefficient of useful ordering (CUO) is always less than one. For a machine to eliminate more uncertainty (thus to perform a more informative computation) it must first increase its own capacity for uncertainty (e.g., by complicating its program). This is impossible for a device with purely deterministic behavior.

In other words, **a deterministic device cannot eliminate more uncertainty with its computation than was eliminated to perform the computation itself**. Or, a deterministic device cannot increase its own behavioral complexity. By analogy with thermodynamics, one cannot create an isolated, deterministic self-"developing" machine of the first kind.

Modern computers and networks have achieved unprecedented complexity. Models created by generations of engineers, trained on vast datasets, can eliminate uncertainty far beyond the capacity of any single individual.

Nevertheless, we are still dealing with machines that deterministically execute their embedded behaviors. The current architecture fundamentally lacks the capacity for independent development.

So-called "artificial intelligence" eliminates uncertainty by mapping queries to a set of *answers based on pre-existing information in its model (some of which may be false). While useful for trivial tasks, these systems cannot generate fundamentally new information not already present in their training data.*

A programmed automaton or a trained model converts electricity into the work of memory ordering, performing computations whose complexity is determined by its creators' intelligence and its embedded data — that is, by uncertainty already eliminated within the model.

The efficiency of this work is immense, as computers scale up the elimination of uncertainty through deterministic repetition. But a machine, as currently conceived, cannot independently surpass the complexity designed into it. This limitation also applies to humans, who can lose the capacity for retraining and independent information generation with age.

In the 19th century, humanity was obsessed with perpetual motion machines that can generate new energy and the spontaneous generation of life. ***The illusion*** *of the 20th and* ***early 21st centuries has been a religious faith in a deterministic device that can generate new information and the spontaneous generation of consciousness within it.***

1.2 Problem Formulation

The Problem of Determinism's Limits

As existence changes non-deterministically, any deterministic device accumulates errors and internal inconsistencies at the physical level and in its memory states. Ultimately, in the mismatch between its embedded models and computations and the changed existence.

A deterministic device is incapable of increasing its own behavioral complexity. One way to improve its CUO is through optimization: increasing the ratio of "useful" computation to the device's own internal ordering by reducing algorithmic complexity and resource requirements (up to code generation). But optimization does not lead to greater complexity.

A non-deterministic device, meanwhile, has no practical value because it introduces uncertainty rather than eliminating it, leading to an even faster accumulation of errors than its deterministic counterpart.

Is it possible, then, to bypass the limits of determinism using a deterministic device? Can a device be made to solve new, more complex problems autonomously and adapt to a non-deterministically changing existence?

Tasks Definition

By analogy with humans, who (albeit rarely) transcend their own deterministic limitations, we can propose the following tasks to address the problem:

1. **Interaction with reality**, determined by the device's current state, to generate information (eliminate uncertainty) from input data.
2. **Deterministic identification of errors** and inconsistencies between the device's current state and reality, based on information from step 1 and embedded criteria.
3. **Non-deterministic complication** or increase in the uncertainty of the device's own state (its program, ontology or parametric model).
4. **Non-deterministic generation of hypotheses** (additional states) that resolve the uncertainty introduced in step 3 (e.g., through independent programming or learning).
5. **Deterministic verification of hypotheses** from step 4 by re-evaluating them against information from steps 1 and 2 to resolve the identified inconsistencies.
6. **Deterministic transition to a new state** by discarding unverified hypotheses.

This list may seem to reduce the problem to only two novel tasks (3 and 4), as the others can be already addressed with varying success and CUO. However, it actually introduces deeper uncertainty, raising new questions of which some have remained unresolved for centuries:

1. How can a deterministic device's state uncertainty be increased autonomously and dynamically (complicate its algorithms without recompilation, self-expand its ontology or perform fully unsupervised data collection and training)?
2. How are hypotheses generated that are not determined by the current model and state (i.e., not limited by the current ontology)?
3. What are the criteria for truth and verification?
4. How complete can a formal representation or model of reality be within a device's states and in principle?

Therefore, solving tasks 3 and 4 requires proposing formal answers to these fundamental questions.

Definition of (Meta)ontology

Returning to the issue of device determinism, it is worth noting a principle or approach in computer science that still manifests even in the von Neumann architecture: the separation of program and data. More broadly, this is the separation of data from the real world from its virtual or ideal representations (methods, functions, matrix models) and entities (structures, classes).

This separation makes it impossible to solve the first question raised in the previous section. Currently, the author is unaware of any formalisms where a system's behavior (its entities and mappings) is described as dynamically as its data. That is, where they are described at the same level of abstraction and used dynamically by the device (even with low CUO) without recompilation or retraining.

On a more fundamental level, such a separation exists in all ontologies known to the author: as the division of the model from reality, physics from metaphysics, the real from the ideal, and so on. This division is either postulated *ad hoc* in each field without explaining the general mechanisms of their interaction, or it is simply denied without recourse (for example, by various forms of materialism).

Thus, information, as a metaphysical phenomenon of ordering, is contained and transmitted in computational devices and over communication channels as an abstraction over physical flows of electricity (along conductor paths), which change the states (charges) of material cells of physical memory. Software is not objectively registered by physical instruments and does not manifest in existence outside of hardware, and any interpretation of the physical state of output devices is subjective.

An honest objectivist must either raise the question of reducing computer science exclusively to the study of electrical activity in semiconductors and statistical research on pixel color distributions on screens and pin signals, or admit that objectivism itself is just one method of cognition, subjectively limiting its toolkit — and thus its cognitive boundaries — to material objects.

The ontological status of software, natural laws, and mathematics is typically an unspoken assumption in science. A physical law, model, or program that has experimental confirmation and/or practical application is simply accepted and used as is. Attempts to discuss their origin and the grounds for their existence are considered secondary. The very nature of formalisms as such and their relation to material phenomena are often omitted.

But it is precisely the absence of a level of abstraction higher than the formalisms, models, and laws themselves that makes their dynamic description and use impossible. This is still reflected even in the von Neumann architecture: data from the real world is dynamic, but ontologies, programs, and laws are static (even within the same heap) because the latter are not defined by the machine itself, but by the mind of the human using it during the programming, environment description, or training phase.

Therefore, a necessary condition for solving the problem of a dynamic ontology is the description of some (meta)ontology as a formalism that:

1. Contains its own foundations, which are no less abstract than any ontology.
2. Non-contradictorily describes and operates with various ontologies and worldviews.
3. Describes the same phenomena at different levels of abstraction.
4. Explains the nature of formalisms as such and the limits of their application.

From a philosophical perspective, the term "metaontology" is a pun. Ontology as the study of existence or being is already at a meta-level relative to any specific formalism. However, in computer science, an ontology is often understood as a local collection of entities, rules, and methods within a specific domain.

2. Formalization of Monads

2.0 The Periodic Table of Phenomena

Since the ontologies used by humanity are created by humans, human experience itself serves as their common highest-level abstraction. A more general and abstract foundation for these ontologies, in turn, lies beyond the limits of human experience.

Human experience is formalized or described in a multitude of ontologies. Due to its vastness, a single individual typically has access to only their subset, which leads to the denial of other ontologies and, consequently, to ontological contradictions and mutual refutations. This generalization is hindered by a general inexperience with formalization, a trait prevalent only in scientific worldview.

But even within physical phenomena, one can differentiate levels of abstraction. For example, electromagnetism, formally discovered in the 19th century, is fundamentally different from the mechanics of macroscopic bodies. Despite various views and critical attitudes toward interpretations of quantum mechanics, their undeniable achievement was drawing widespread attention to the incompleteness of the mechanistic worldview.

No one in their right mind would try to measure electric current with a dynamometer. Nor would they attempt to prove the exclusively corpuscular nature of electricity based on its particle phenomena (which does exist) without also considering the more abstract phenomena of waves and fields (setting aside the adherents of corpuscular force carriers).

Yet when it comes to metaphysical phenomena, the reduction of emotions, thinking, consciousness, etc., exclusively to their physical manifestations (which do exist) is not uncommon. Idealists, on the other hand, have still not systematized or proposed general mechanisms for the "descent of spirit into matter," becoming mired in unverifiable scholasticism. Or appealing to an unknowable physics-metaphysics divide - a position as helpless as that of the materialists.

The table presented below contains a system of phenomena and meta-phenomena from differential phenomenology. They are used subsequently to propose a formal model of (meta)ontology. Its cyclical structure stems from the requirement that it contain its own foundations and from the properties of reality to be postulated below. Each phenomenon, as a cell in the table, may contain several names or aspects.

The rows indicate levels of abstraction. Phenomena in the same row have different natures but are interdependent and interconnected. Immanent to each level of abstraction is a pair of phenomena (in the rightmost column) that manifest on it, unfolding "up" and "down" the levels. The columns show the levels of abstraction for the same phenomena.

	Degree of Freedom	*Potential*	**Realization**	*Capacity*	**Flow**	*State*	***Change***	Contradiction	**Similarity**	
	Absolute Freedom	*Potential* *Mind* *Differentiation* *Infinity* ∞← ↓	***Realization*** ***Reality*** ***Oneness*** ***Monad*** ← ↑ ←∞							
	Essence	***Experience*** ***Idea*** ∞← ↓	**Being** **System** ← ↑	***Capacity*** ***Entity, Set*** ***Law, Plot*** ***Archetype*** ← ↓	**Flow** **Existence** **Energy** **Good** ← ↑ ←∞					
Time	*Causality*	*Process* ∞← ↓	***Society*** ***Personality*** ← ↑	*Consciousness* *Actor* ← ↓	***Intention*** ***Attention*** ***Understanding*** ← ↑	*State* *Inertness* *Variable* ← ↓	***Action, Act*** ***Change*** ***Consequence*** ← ↑ ←∞			
Time	**Rationality**	***Complexity, Entropy*** ∞← ↓	**Individual** ← ↑	***Intelligence*** ← ↓	**Thought** **Computation** **Function** ← ↑	***Number*** ***Ratio*** ***Data*** ← ↓	**Ordering** **Truth** ← ↑	***Contradiction*** ***Error*** ← ↓	**Similarity** **Recursion** ← ↑←∞	
Space	*Image* *Form*	*Beauty* *Attraction* ∞← ↓	***Life*** ***Organism*** ← ↑	*Psyche* *Soul* ← ↓	***Emotion*** ***Reflex*** ← ↑	*Symbol* *Memory* ← ↓	***Perception*** ***Somatics*** ← ↑	← ↓	← ↑	… ←∞
Space	**Electricity**	***Voltage*** ***Field*** ∞← ↓	**Body** **Device** ← ↑	***Circuit*** ***Plane*** ← ↓	**Radiation** ← ↑	***Charge*** ← ↓	**Current** **Heat** ← ↑	← ↓	← ↑	… ←∞
Space	*Matter*	*Distance* ∞← ↓	***Particle*** ← ↑	*Vector* *Line* ← ↓	***Force*** ← ↑	*Mass* ← ↓	***Motion*** ***Impulse*** ← ↑	← ↓	← ↑	… ←∞

2.1 Definition and Quantifiers

Differentiation and Oneness (Realization)

Let us differentiate (postulate) one primordial pair of phenomena: **differentiation / potential / mind / infinity** and **oneness / realization / reality / monad**.

Oneness of differentiated / potential realization / minded reality / finite-in-itself monad is what we differentiate as **knowledge**. Both the difference and the oneness of monads is itself a monad. This infinite, recursive self-definition is elaborated upon below and summarized at the end of the chapter.

Let us differentiate (define) the formal operator of oneness / realization / reality / monad (…, …) or … : … as well as the absolutely unrealized monad (), which formally represents the least-realized potential, undifferentiated oneness or the **unknown**.

Let us differentiate one pair of **transcendent phenomena** or meta-phenomena: **absolute differentiation** / potential / mind / infinity ∞ and **absolute oneness** / realization / reality / monad (∞).

Set and Existence

The absolute potential differentiates itself in potential differences which are called **experiences** / ideas. Undifferentiated, continuous **existence** / flow / energy / good is differentiated as the phenomenon that realizes or manifests these experiences or ideas.

That is, experiences (ideas) are differentiated as finite (discrete) potential differences from the undifferentiated continuous potential mind / infinity. They are realized and known through the flow of undifferentiated continuous existence / energy in discrete and distinct **entities** and manifestations.

Entities outside of time and space are not known in-themselves or as such. Let us differentiate the **set** / entity of a realized experience $@A$ and the operator of set / manifestation / existence / belonging @ such that:

1. $() @ \infty$
   (The empty or unmanifested set as an absolutely unrealized monad / reality manifests absolute potential; i.e., it is absolutely potential and unknown)
2. $() @ ()$
   (The empty or unmanifested set, as an absolutely unrealized monad, manifests itself; i.e., it is one as-such or in-itself as the unknown)
3. $@A$: $[@A_1, @A_2, ..., @A_n]$ или $[@A_1, @A_2, ..., @A_n]@A$ или $(@A, @A_1, @A_2, ..., @A_n)$
   (The experience / idea A of a non-empty or manifested set / entity $@A$ can be realized and known in a monad as the oneness of the different monads of entities $@A_i$ of the one entity $@A$).

Let us differentiate (define) the quantifier of entity $\forall$ such that for a set or entity $@A$, different into a set of entities $[@A_i]$:

$\forall [@A_i] : @A$ or $(@A, \forall [@A_i])$
(*Each different monad of the set / entity* $[@A_i]$ *manifests the one entity / set* $@A$).

Due to the infinity of the absolute potential / mind, its finite experiences / ideas have no end. Entities and their existence are finite, discrete, or quantized in the realization of experience. The potential multiplicity of experiences and the flow of existence or energy, however, are in themselves continuous and have no count or limit (this relates to the main paradoxes of set theory, theoretical physics, theology, etc.).

Outside of time and space, entities can be endlessly differentiated and realized into sets of their differences. The monad $@A$ as an entity or set of entities is also called a capacity / law / archetype.

The operation of a set or entity $@A$ indicates an archetype in essence; its formal result will be a set, archetype, or capacity of monads (both as entities and objects in time and space) that manifest the archetype or belong to the set or capacity of experience / idea A. The entity quantifier ∀, in turn, points to the monad of an entity manifested in the entire set or in each monad of the entities of the set.

The archetype $@Node$ is differentiated as the entity or archetype of the monad as-such. The archetype $@Archetype@Node$ is differentiated as the entity / archetype of the archetype / entity as-such. Thus, $@Archetype@Node$, $@Node@Archetype$ — the archetype / entity of the monad as-such is an archetype / entity, and the archetype / entity as-such is a monad.

The monads $@A_i$ that manifest the entity / archetype $@A$ or belong to the set $@A$ (still outside time and space) can be further different as archetypes. Thus, entities form a hierarchy of differences from more abstract to more cognized entities outside of time and space.

Thus, the monad / reality can be differentiated in its existence as:
$(@id,\ (@Archetype))@Node$, where
$@Archetype$: $[@A_i@Archetype@Node]$ is the set of monads of entities.

2.2 difference of States and Changes

State and Change (Action)

Entities are differentiated outside of time and space in-themselves or as such, and therefore cannot become one and be known by-themselves or as such. The less abstract (mindfull) potential of their oneness is differentiated within time as a process.

Through the realization of potential differences as processes, a single entity can manifest in time as a set of monads. These temporal manifestations are differentiated as **states**, inertias, objects, or variables. The **change** or action of states / inertias is differentiated over time.

Each object, inertia, or variable can, in turn, be differentiated in various states in time, but does not possess its own state outside of time. Change itself is continuous, but it becomes observable through the discrete states of these inertias and can therefore be described at a less abstract level through the states of discrete inertias differentiated in monads as "past" and "future," or "before" and "after" a change.

Every change or action is unique in time and cannot be formalized either "before" (non-determinism) or "during" its realization (since it is recorded only "after"), but all changes are conditioned by the entities that the changing objects manifest. Outside of time, at the essential level of abstraction, one can describe classes, sets, or entities of changes (which allow entities oneness). These are called **scenarios**, plots, or laws (including natural laws).

Formally differentiated (defined) are the archetype $@Variable$ as the entity of monads representing variables or states, the archetype $@Action$ as the entity of monads representing action/change. And the archetype $@Plot$ as the entity of monads of scenarios, in which changes are discretely differentiated at the essential level as sets of inertias or variables possessing "before" and "after" states. They will be differentiated and formally described later, taking into account less abstract rational and pre-rational attributes of the monad.

For the formal difference of a monad of an object realizing experience / idea $\#e$ in time from the entity $@E$ outside of time, the identification operation # or the quantifier of finiteness / existence in time ∃ is used:
$\exists\ e$ or $\#e$

The entity quantifier for a set of objects ∀ differentiates in the entire set of objects $[\#a_i]$ the entity $@A$ as the entity differentiated in all its objects:
$\forall\ [\#a_i] : @A$ or $(@A,\ \forall\ [\#a_i])$

The entity quantifier for a set of objects is not equivalent to the quantifier for a set of entities as-such. This is due to the emergent or systemic properties that arise when entities are instantiated as objects, a concept that will be discussed later.

The monad / reality is formally differentiated in time as:
$(\#id,\ (@Archetype),\ (@Action))$, where
$@Action$: $[action_i @Action]$ is the set of monads of actions/changes.

Number and Ordering

While different entities' objects can change arbitrarily in time and realize processes, they cannot achieve oneness through change alone. The less abstract (mindfull) potential of their oneness is differentiated in time (though not yet in space) as **entropy**, complexity, chaos, or uncertainty.

The multiplicity of an entity's manifestations over time allows its objects to realize different processes and change in different ways. Which is differentiated as the phenomenon of **contradiction**. When objects are **ordered**, they cease to change relative to each other, which is different as **similarity**. The potential of uncertainty lies in the unknown of how objects (e.g., the states of a device's memory cells or particles of a substance) will be ordered "before" the processes are realized.

The progressive ordering of discrete objects in time is differentiated and formalized as the natural number series. When contradictions in the states or inertias of unified objects are resolved, their states are brought into relation with one another. Such relations of natural sequences are formalized from experience as **rational numbers**. (Supra-rational or irrational numbers can be understood as a projection of more abstract, essential phenomena into the numbers).

Number is a less abstract manifestation of the phenomenon of inertia or state that arises when uncertainty is eliminated or inertias are ordered. Ordering, as an action of uncertainty elimination that changes the numerical state, is a less abstract manifestation of the phenomenon of change or action as such.

Besides their value, both natural and rational numbers are defined by entities of relation, such as size, mass, cost, relative speed of a process's realization projecting abstract time potential into measurable physical 'time' etc. They are also differentiated through discrete standards of relation or units of measurement (or they are dimensionless if the standard is related to itself).

Formally differentiated (defined) are the archetype $@Attribute$ as the formal entity of states or intertias, the archetype $@Float$ as a floating-point rational number, the archetype $@Unit$ of units of measurement monads, and the archetype $@Ratio@Attribute$ of monads of numerical or rational attributes. These attributes themselves are differentiated as oneness of a rational number, a unit of measurement, and the monad with which the relation is made.

The monad / reality is formally differentiated with its rational numerical states as: $(\#id,\ (@Archetype),\ (@Intention),\ (@Ratio))$,, where $@Ratio\colon\ [(v_i@Float,\ u_i@Unit,\ t_i@Node)@Ratio]$ is the set of monads of the numerical attributes of the monad, in which the states or consequences of its relations with other monads or with itself are different.

Memory and Perception

Different orderings may be rationally correlated and ordered in time, but not yet ordered as oneness. The less abstract potential for their oneness in space is differentiated as **beauty** perceived in three-dimensional forms and images.

In physiology, perception is based on what is known as the first signal system. In our context, we extend this concept to include the raw **symbols** of natural language (both hieroglyphic and phonetic) prior to any syntactic or lexical abstraction. By stimulating receptors and more complex sensory organs, these signals change the state of a living system at a pre-rational level, one less abstract than number. Such signals are what we define as **symbols** or memories.

In modern computational devices, such memories are represented as text characters in various encodings (including emojis), images, audio, video, and other media files.

Formally differentiated (defined) are the archetype $@File$ for monads containing memories in binary form, the archetype $@Memory@Attribute$ for monads of non-numerical or pre-rational attributes. A pre-rational attribute is therefore defined as a oneness of the file monad (the memory itself) and the target monad that was the source of the perception.

The monad / reality is formally differentiated with its pre-rational, non-numerical states as: $(\#id,\ (@Archetype),\ (@Intention),\ (@Ratio),\ (@Memory))$, where $@Memory\colon\ [(f_i@File,\ t_i@Node)@Memory]$ is the set of the monad's non-numerical attributes, in which the states or consequences of the monad's perception of other monads (or itself) are recorded.

Material Body

Monads differentiated through forms and images can perceive each other in space but still cannot form oneness. A less abstract (mindfull) potential of their oneness in space is differentiated as electrical **voltage** or field.

The manifestations of the phenomena of inertia and change here appear as **charge** and electric **current** / heat. This level of abstraction will not be used in the formalization of monads for the time being.

Monads differentiated in fields can inter-act via electromagnetism, but still cannot unify. A less abstract potential is differentiated as the one-dimensional potential of **distance**, which is realized in particles of matter. The manifestations of the phenomena of inertia and change in matter appear as **mass** and **motion**.

In matter, the conceptual "octave" completes its cycle, and the meta-phenomenon of reality as such manifests. This occurs through the oneness of two aspects: the flow of energy condensed to a gamma-range EM wave as **force** and the capacity of a one-dimensional vector. This oneness realizes or "collapses" into a standing wave of energy/existence, forming a discrete **particle**.

A materialized monad as a particle acquires a location in space. This location is defined by coordinates as differentiated distances (or linear potential differences) relative to other particles or points of reference. We rationally measure these relative to discrete units of length, down to the quantum of linear capacity or the boundary of discrete existence, the Planck constant.

Thus, the mutual arrangement of particles brings into reality the least abstract linear phenomena: **vector** as linear capacity / entity and **force** as linear flow / existence/ energy. The ordering of particles manifests in increasingly heavy elementary particles.

Materialized monad-particles represent the realization of the least abstract (mindfull) entity in time and space. Therefore they form the basis for the process of oneness which is reverse to differentiation. The oneness of monads of different and increasingly abstract (mindfull) entities is called evolution in the naturalistic worldview. The realization of a monad as a particle (or a oneness of particles) is also termed **realization-in-being** (or more simply, **materialization**).

Particles that realize electrical potentials or fields with non-zero temperature and charge are capable of ordering themselves via bonds into atomic structures and chemical compounds, and further into physical **bodies**. They also manifest — bringing them from non-being into reality — the more abstract, planar phenomena of capacity and flow: the electrical **circuit** and EM **waves**.

The monad / reality is formally differentiated with its material aspects as:
$(\#id, (@Archetype), (@Intention), (@Ratio), (@Memory), (@Mass, @Disposition))$, where
$@Mass@Ratio$ is the set of numerical attributes of mass(es),
$@Disposition$: $[disp_i@Ratio]$ is the set of numerical attributes of the relative positions of the monad's bodies.

2.3 Realization of Capacities and Flows

Life, Psyche, Perception

This level of abstraction marks the difference between physics and metaphysics. Metaphysical phenomena, being more abstract (or mindful), are always realized upon a physical basis, yet they are not reducible to it. This irreducibility is due to their additional degrees of freedom or levels of abstraction.

By their very essence, these phenomena are subjective, even at the level of imagistic perception. Consequently, as suggested earlier, they fall outside the domain of science — at least from the viewpoint of objectivism, which is constrained by its own self-imposed methodological limitations.

Monads of physical bodies, realized as one in space through elementary and electrochemical bonds, are capable of realizing as one at a more abstract (mindful) level as more abstract potentials are realized. The realization of the potential of **beauty** or attraction, as the most abstract or mindful potential in three-dimensional space, is manifested in the monads of **living** organisms.

The realization-in-being of a material body and the phenomena of memory and perception gives materialization to a more abstract pair of capacity and flow phenomena. The **psyche** (soul), which serves as an imagistic capacity as an abstraction above the physical structure of cells and organelles (the genotype as memory of the species' form), tools, and works of art (the phenotype as collective memory of the species / cultural code of an ethnos / culture of a society). On the other hand, it manifests as **emotions** / reflexes flows, which are immanent expressions of energy in living matter, abstract above the physical flows of mechanical forces and EM radiation.

A reflex or emotion, as an undifferentiated (differentiable) flow, is not realized in discrete being without a discrete (quantized) capacity. In living matter, this capacity is the psyche (soul), acting as an abstraction (or mindfulness) over the physical organism. The psyche contains the monad's memory state and determines its reflexive behavior based on both conditioned (learned) memory and inherited (instinctual) memory from the species, ethnos, and culture of previously realized and perceived experience.

The emotional aspect of the organic manifestations of energy is not yet represented in computer science, but reflexive behavior has been modeled for over half a century by means of the perceptron and/or their systems (neural networks). These reproduce the conditioned reflex through parametrization (training, conditioning) of a linear or matrix operator with error minimization (positive/negative reinforcement) according to a set of input and output data from a training sample (stimuli and reactions).

Let us define the Archetype $@Reflex@Plot$ as the monad of a reflexive scenario, which unifies the variables $Arg@Variable$ for the argument "before" and $Res@Variable@Memory$ for the non-numerical result "after" the reflex is realized. One can represent a reflex as a mapping of an arbitrary monad (with numerical and/or non-numerical attributes) to a memory monad (of symbols or signals of the first system), realized, among other ways, by means of a neural network monad as a system of perceptron monads.

The monad / reality is formally differentiated with its psyche and reflexes as:
$(\#id,\ (@Archetype),\ (@Intention),\ (@Ratio),$
$(@Memory,\ @Reflex),\ (@Mass,\ @Disposition))$, where
$@Reflex$: $[\ (Arg_i@Variable,\ Res_i@Variable@Memory)@Reflex\ ]$ is the set of monads of reflexes.

Individuality, Intellect, and Thought (Correlation)

This level of abstraction marks the difference between space ("the sensible") and time ("the intelligible").

Realized as one in space through reflexive connections and perceptions, the monads of living organisms can then realize at a more abstract (mindful) level as higher potentials are realized. The realization of this potential for complexity or uncertainty leads to the emergence of the monad of the **individual**.

Realization-in-being of a living organism and the phenomena of relation (number) and ordering materializes a more abstract pair of capacity and flow. This pair consists of **intellect** (mind) as a rational capacity, and **thought** (or correlation/computation) as an immanent manifestation of energy in intelligent life.

This rational capacity, the intellect, contains the monad's formal or numerical state. It governs rational thought and the ordering of reality based on previously acquired information, such as relations, numbers, and their arrangement into models and ontologies.

Rational flows of correlation or computation, and rational capacities as formal models and their aggregates, formalize various logics, mathematics, and the rigorous disciplines that use them.

Formally differentiated (defined) are the archetype $@Function@Plot$ as the monad for a function or computation scenario as oneness of the variables $Arg@Variable$ for the argument "before" and $Res@Variable@Ratio$ for the numerical result "after" the computation is realized. A function can be represented as a mapping of an arbitrary monad (with numerical and/or non-numerical attributes) to a numerical attribute monad, including by means of a mathematical function.

The monad / reality is formally differentiated with its intelligence and computations as:
$(\#id,\ (@Archetype),\ (@Intention),\ (@Ratio,\ @Function),$
$(@Memory,\ @Reflex), (@Mass,\ @Disposition))$, where
$@Function\colon\ [\ (Arg_i@Variable,\ Res_i@Variable@Ratio)@Function\ ]$ is the set of monads of functions.

Society, Consciousness, and Intention

This level of abstraction marks the boundary of rationality and formal cognition. Supra-rational phenomena, being more abstract and multi-dimensional, do not project completely onto rationality. They are therefore often denied by intelligent life that has not sufficiently realized the phenomena of consciousness and state as-such (will).

A formal description of process dynamics, entities-in-themselves, and absolute meta-phenomena is never complete; there exists an infinite (potential, non-realizable within rationality) set of non-equivalent ways to describe them. Therefore, the formalization of supra-rational phenomena presented here is knowingly incomplete and non-unique.

Static, pre-rational phenomena can be fully and consistently contained within the rational. For example, there are many equivalent ways to describe a system's physical state using different formalisms of classical mechanics and thermodynamics, different media file formats, or different natural and formal languages.

However, the dynamics of a system, in the general case and over a sufficiently long period (time potential), cannot be fully formalized due to non-determinism and irreversibility. In the 21st century, de-sacralizing formal rational cognition — and rejecting it as the supposedly highest level of abstraction (mindfulness) of experience realization — is as difficult as was the departure from a pre-rational, scholastic-sensory worldview during the Enlightenment.

Monads of intelligent life, realized as one in time through ordered connections and correlations, can realize as one at a more abstract level as higher level potentials are realized. The realization of the potential of process manifests in the monads of **societies** and personalities (collective consciousness as oneness of individuals and personal consciousness as oneness of individual sub-personalities with differing ontologies).

The realization-in-being of intelligent life and the phenomena of state and action materializes a more abstract pair of capacity and flow: **consciousness** (the subject) as a causal capacity, and **intention** (or understanding/observation) as an immanent manifestation of energy in conscious individuals.

This causal capacity, consciousness, contains the state or inertia (will) of the monad as-such and discretely actualizes its continuous and undifferentiated actions over time. These causal flows of intention and capacities of will cannot be fully formalized.

During the realization of a process, the subject and object (or multiple subjects, if several monads possess consciousness and will within one process) become one. The scenario of the process is therefore realized and cognized within the monad of both the subject and the object.

Monads have access to an infinite (potential, unformalizable) set of action scenarios in time. A rational projection or incomplete formal representation is an enumeration of possible scenarios in which the monad's actions are differentiated and cognized in their essence.

The subjective aspect of change can be seen as the set of transitions of the subject's attention or intention toward various monads. The objective aspect can be seen as a set of cause-and-effect relationships or "triggers" — actions that the monad performs, or that are performed upon it, under certain conditions.

A rational projection of these cause-and-effect relationships is a conditional function, which defines the conditions (or probability) for a given scenario's realization.

Thus, the archetype $@Action$ can be further differentiated (defined) as a monad representing the entity of change. It realizes as one a conditional function ($@Function$) with a target monad ($@Node$ or $@Plot$ for a scenario), to which the actor's intention is directed, and which is executed if the action's condition is met.

Formally differentiated (defined) is the archetype $@Intention$ as the entity for monads of intentions, which realize as one the monads of states differentiated in time: $Before@Variable$ and $After@Variable$.

The monad / reality is formally differentiated with its consciousness and intentions as:
$(\#id,\ (@Archetype),\ (@Action,\ @Intention),\ (@Ratio,\ @Function),$
$(@Memory,\ @Reflex), (@Mass,\ @Disposition))$, where
$@Action$: $[(target_i@Node,\ condition_i@Function)@Action]$ is the set of monads of actions,
$@Intention$: $[(Before_i@Variable,\ After_i@Variable)@Intention]$ is the set of monads of intentions.

System (Being)

Realized as one in time through intentions and actions, the monads of societies and personalities can realize as one at an even more abstract (or mindful) level as higher level potentials are realized. The realization of the potential of **experience** as such manifests at the essential level in the monads of **systems**, or being as such.

Consider the experience of an entity $@A$, which is itself differentiated into entities $@B@A$ and $@C@A$. These can be further differentiated outside of time and space into more granular experiences, realized in sets of entities like $@B_i$ and $@C_i$. While different realizations of $@B_i$ and $@C_i$ are not reducible to the experience of A, their realization as one in time and space leads to the cognition of the more abstract entity $@A$.

In this process, entity $@A$ realizes or cognizes within itself new properties that are not found in entities $@B$ and $@C$ alone. This phenomenon, immanent to entities, is called a systemic property or emergence. It arises because the abstraction (mindfulness), differentiation or potential difference of the entire experience of $@A$ is greater than that of its constituent experiences$@B$ and $@C$ (which, in turn, have a greater potential difference than their own differences, $@B_i$ and $@C_i$, and so on for the entire entities' hierarchy).

While a set is a monad composed of differentiations of a single entity, a system can be represented as a monad of disjunctions — a oneness of monads and sets of different entities. Through its existence, this system realizes-in-being a more abstract (or mindful) entity than any of its constituent parts.

A necessary condition for system-ness or emergence is the realization-in-being, or materialization, of the system. Entities do not realize as one as-such without cognizing themselves as oneness of less abstract phenomena in time and space (herein lies the "function" or "design" of being, which is sometimes not differentiated in the experience of idealists and theorists).

Thus, the most abstract (mindful) realization of experience available to humans in being is the realization-in-being or materialization of systems in time and space, and the cognition within them of entities outside time and space, such as sets of archetypes, classes, species, patterns, and scenarios or laws, including natural ones.

Cognition as-such is more abstract (mindful) than rational cognition due to non-determinism, but it is limitedly projected into rationality over time in an infinite (potential, non-realizable within rationality) set of incomplete (including this one) and non-equivalent models, theories, teachings, and ontologies.

Formally differentiated (defined) are the archetype of a system or being $@System$, and the formal operator of a system, emergence, or disjunction of entities/sets in being $\&$.

The monad / reality is formally differentiated with its systemic properties as:
$(\#id,\ (@System,\ @Archetype),\ (@Action,\ @Intention),\ (@Ratio,\ @Function),$
$(@Memory,\ @Reflex), (@Mass,\ @Disposition))$, where
$@System\colon\ [@A_1\ \&\ @A_2\ \&\ ...\ \&\ @A_n]$ is a set of (sub)systems as a realization-in-being or materialization of monads of different entities,
The remaining attributes and entities of the monad are its systemic properties, distinct from or not reducible to the entities and attributes of its (sub)systems.

From the differentiation (definition) of entities and the operator of a set/entity ∀, it follows that any entity in the hierarchy of entities must ultimately manifest some most abstract (mindful) "absolute entity," from which all experiences are cognized in the hierarchy of entities:
$\forall\ [@E] : @\infty$?

If we imagine the absolute potential as an infinite (circular) line, then there exists an infinite number of differentiations within it of finite intervals of experience. For each such difference, the complete or partial onenesses of intervals of experience through being as systemic properties generate ever new sets of differences. Then the "absolute entity" would be the complete interval as the systemic property of all potential differences of experiences.

Such an entity cannot be cognized in-itself or as such as a finite or discrete capacity, since entities do not realize as one without realization-in-being or materialization, and their multiplicity is infinite.

But as they cognize themselves in time and space, $\boldsymbol{\mathcal{R}}$ is differentiated as the monad of being or a system that discreetly realizes-in-being a set of the most abstract/rational entities $@E : [@E_i]$, such that:

$\forall\ [@E_1,\ @E_2,\ ...,\ @E_{n-1}] : @E_n\colon\ \exists\ \boldsymbol{\mathcal{R}}$ or $(\#\boldsymbol{\mathcal{R}},\ [@E_1\ \&\ @E_2\ \&\ ...\ \&\ @E_{n-1}\ ]@System,\ @E_n,\ ...)$
(The monad of being materializes as a system the most abstract / mindful entity as the realization of the greatest potential difference of experience)

But the realization as one of entities in time and space, that is, in being or a system, leads to emergence or the realization of ever new systemic properties, new entities, and the cognition of new experiences, not reducible to the experiences of already realized as one (sub)systems:
$\forall\ [@E_1, @E_2\ ,\ ...\ @E_n,\ ...\ \infty] : (\#\boldsymbol{\mathcal{R}}, [@E_1\ \&\ @E_2\ \&\ ...\ \&\ @E_n\ \&\ ...\ \infty]@System,\ ...\ \infty)$

That is, the cognition of the "absolute entity" in being $\boldsymbol{\mathcal{R}}$ as the entity of the most abstract / mindful finite experience has no end in the infinite set of differentiations of finite experiences. Being manifests in finitude/existence a one but infinite (absolutely potential, not realizable in existence) set of cognitions of experiences, or infinitely realizes the absolute potential as absolute reality (∞).

Thus, the infinite but discrete differentiation (definition) of the monad / reality through recursion reflects the immanent oneness in reality of infinity and finitude, differentiability and discreteness, potentiation and realization. It is manifested in infinite recursive self-cognition through the realization of differences in the potential of experience, time (processes, complexity), and space (beauty, voltage and distance) immanent to the absolute potential, in the oneness of differentiated (defined) phenomena.

Absolute reality (∞) cannot be cognized as-such, but is cognized through being $\boldsymbol{\mathcal{R}}$ as the oneness of being $\boldsymbol{\mathcal{R}}$ and the absolute potential ∞:
$(\boldsymbol{\mathcal{R}}, \infty)$.

(Absolute reality is cognized as the oneness of absolute potential and its realization-in-being, or as the infinite self-cognition of the infinite mind in being).

2.4 Epistemology of Monads

Data and Judgment

Data is defined as a monad or a oneness of phenomena of state or inertia that are no more abstract than numbers. That is, it is a oneness of relations within symbols and memories (including reflexive models), and of charges and masses (the physical states of objects).

For example, data in the form of natural language text or mathematical formulas is a oneness of the texts' and formulas' symbols and the relations of the language's grammar or the mathematical operators. The phenomenon of data is rational, meaning it is realized through the capacity of the intellect.

Judgment is defined as the oneness of the phenomena of data and the flow of intention (e.g., assertion, questioning, indication). It is therefore causal, supra-rational, and realized through the capacity of consciousness.

Entities as-such or in-themselves are more abstract than judgments about them. This means they do not project completely and unambiguously into data, but rather into an infinite set of datasets about them. To work with them formally, a subjective projection of the entity into an object is possible, based on common properties identified from the set of objects that manifest it.

$\forall\ [\#(a_i,\ @Attr@Attribute)@A] :\ (\#A,\ @Attr)$

Such a projection of an entity into an object is called an **a priori judgment** or a judgment about the entity. The further projection of an a priori judgment about an entity $@A$ into data about it as an object $\#A$ is called **a priori data** about the entity $@A$, or the object of entity $@A$. The analogue of a priori data in computer science are entities, classes, and data structures.

A Priori and A Posteriori Data

A posteriori or "empirical" data is defined as data realized from experience at a level of abstraction lower than rationality, i.e., in space. It is the oneness of the realization of pre-rational "sensory" experience as from memories of perceived images and the physical states of objects and the relations ordered from them in the mind.

A posteriori data allows one to make a priori judgments about the entities of sets of perceived and correlated objects.

A priori data is defined as data realized from experience at a level of abstraction higher than rationality, i.e., in time and in essence. It arises from an a priori judgment as the oneness of conscious intention and the projection of that a priori judgment about an entity into data.

For example, a posteriori data of an object's coordinates is realized through its perception (not always intentional), the correlation of its location, and its ordering within some model of space with a point (object) of reference. A priori data of an object's coordinates in time (future or past) is intentionally (subjectively and arbitrarily) calculated from a priori judgments about the laws of its motion.

In both cases, data is different in time with the realization of processes. A posteriori data arises as a consequence (a state) of the realization of external processes in space, their perception, and their ordering into data. A priori data arises within the framework of the subject's intentional realization of processes in time and is based on a priori judgments about already cognized entities (sets of laws, classes, archetypes, etc.).

The phenomenon of ordering data — eliminating uncertainty based on perceived memories and the physical states of objects — is also called **truth** (verification, "truth-ification," or justification). Thus, true data is defined as a posteriori data that has been differentiated in space. A priori data is not, in the general case, true until it is realized-in-being as a posteriori data.

Since systemic properties are only realized through the realization-in-being or materialization of a system, they can only be identified from a posteriori data, even if as a priori judgments. A posteriori data is the sole criterion for verifying a priori judgments as true.

However, to materialize systems of ever more abstract entities in being, it is necessary to first realize subsystems of already cognized entities, which is impossible without a priori judgments about them. A priori judgments are also differentiated directly from the realization of supra-rational experience in the form of hypotheses about entities, even before their systemic properties are identified a posteriori. A priori data makes cognition as such possible.

Thus, both a priori and a posteriori data represent interdependent, mutually conditioned, and equally significant aspects of the single phenomenon of cognition, different at different levels of abstraction (mindfulness).

From this proposed differentiation (definition) of truth as a posteriori data, it follows that **truth is local in space and time within the experience of the cognizing monad**.

Data transmitted through communication channels is a projection into data of the sender's a priori judgment and, in the general case, is not true for the receiving monad until it is verified by the receiver as a posteriori.

Thus, "absolute truth" is not realizable within human experience. It is realized in the oneness of all a posteriori cognitions of itself in space and time by the absolute potential (infinite mind).

3. Algebra of Monads

3.1 Operations of Differentiation

Difference As Such ("is different from")

Based on the differentiation (definition) of the meta-phenomenon of differentiation, the difference of monads as such … ÷ … is realized in a monad as the difference of its identifier or realized experience / idea:

1. The difference of one monad from another monad realizing a different experience / idea is the first monad itself.
   $@A_1 \div @A_2 : @A_1$ или $(@A_1 \div @A_2,\ @A_1)$
   $\#m_1 \div \#m_2 : \#m_1$ или $(\#m_1 \div \#m_2,\ \#m_1)$

2. The difference of a monad from itself is an empty monad, or it is not cognized:
   $@A \div @A$: () или $(@A \div @A, ())$
   $\#m \div \#m$: () или $(\#m \div \#m, ())$

3. The difference from the unknown or infinity is unknown or infinite:
   $() \div \#m_1$: () или $(() \div \#m_1, ())$
   $\#m_1 \div ()$: () или $(\#m_1 \div (), ())$

Difference as such … ÷ … is the inverse of oneness as such … : …, but it is not a formal inverse due to systemic properties or emergence.

Difference by Essence ("is not")

The difference by essence between an object/entity monad and an entity monad is defined as:

$(\#m_1,\ @A_1, @A_2)\ /\ @A_2 :\ (@A_1)$
("is not an", the difference by essence is cognized)

$(\#m_1,\ @A_1)\ /\ @A_1 :\ ()$
("is an", the difference by essence is not cognized, they are one in essence)

Similarly, the difference by essence between object monads is:

$(\#m_1,\ @A_1, @A_2)\ /\ (\#m_2,\ @A_2) :\ (@A_1)$

$(\#m_1,\ @A_1)\ /\ (\#m_2,\ @A_1) :\ ()$

The inverse operation, oneness by essence `is`, of an object/entity monad and an entity monad is defined as:

$\#m_1 @A_1\ is\ @A_2 :\ ()$

$\#m_1 @A_1\ is\ @A_1 :\ (@A_1)$

$(\#m_1,\ @A_1, @A_2)\ is\ (\#m_2,\ @A_2) :\ ()$

$(\#m_1,\ @A_1)\ /\ (\#m_2,\ @A_1) :\ (@A_1)$

Difference by Relation ("is not equal to")

The difference by relation between monads of numerical attributes $@Ratio$ is differentiated (defined) as the oneness of their difference by essence and the differences by essence of their numerical values, units of measurement, and targets of relation:

$(\#rat_1, @Attr_1@Ratio,\ v_1@Float,\ u_1@Unit,\ t_1@Node)$

$(\#rat_2, @Attr_2@Ratio,\ v_2@Float,\ u_2@Unit,\ t_2@Node)$

$\#rat_1\ -\ \#rat_2 : ((@Attr_1\ /\ @Attr_2),\ (v_1\ /\ v_2),\ (u_1\ /\ u_2),\ (t_1\ /\ t_2))$

The difference by relation between monads of non-numerical attributes $@Memory$ is differentiated (defined) as the unity of their difference by essence and the differences by essence of their files and targets of recollection:

$(\#mem_1, @Attr_1@Memory,\ f_1@File,\ t_1@Node)$

$(\#mem_2, @Attr_2@Memory,\ f_2@File,\ t_2@Node)$

$\#mem_1\ -\ \#mem_2 : ((@Attr_1\ /\ @Attr_2),\ (f_1\ /\ f_2),\ (t_1\ /\ t_2))$

The difference by relation - between object monads $\#m_1\ and\ \#m_2$ is a monad that realizes as one the differences by relation of the attributes of monad $\#m_1$ from the attributes of $\#m_2$, and the difference by essence of the entities of $\#m_1$ from the entities of $\#m_2$.

$(\#m_1,\ @Ratio_1,\ @Memory_1,\ ...)$

$(\#m_2,\ @Ratio_2,\ @Memory_2,\ ...)$

$\#m_1 - \#m_2 : ((\#m_1),\ (\#m_1\ /\ m_2),\ (@Ratio_1\ -\ @Ratio_2),\ (@Memory_1\ -\ @Memory_2)$

3.2 Operations with Entities

Becoming of an Entity ("start being")

The oneness of an object monad $\#m$ and an entity monad $@A$ is differentiated (defined) as a monad representing the oneness of monad $\#m$, entity $@A$, and the difference by essence between the a priori data of entity $\#A$ and the monad $\#m$:

$\#m\ +\ @A : ((\#m),\ @A,\ (\#A\ /\ \#m))$

The result of realizing as one of monad $\#m$ and entity $@A$ is a monad that realizes as one the entity $@A$, all attributes and entities of $\#m$, as well as all attributes from the a priori data of entity $\#A$ that are different by essence from the attributes of $\#m$.

That is, when an object becomes or starts being an entity, it inherits only those a priori or typical attributes of that entity's object form that are not already different within it. Otherwise, the object retains its unique attributes that are different from the a priori data of the entity.

Negation of an Entity ("stop being")

The negation of an entity $@A$ from an object monad $\#m$ is differentiated (defined) as a monad representing the oneness of the difference by relation between monad $\#m$ and the a priori data of entity $\#A$, and the difference by essence between monad $\#m$ and entity $@A$:

$\#m\ -\ @A : ((\#m\ /\ @A),\ (\#m\ -\ \#A))$

The result of negating entity $@A$ from monad $\#m$ is a monad that realizes as one all entities of $\#m$ that are different from $@A$ and all attributes of $\#m$ that are different by relation from the a priori data of entity $\#A$.

That is, when an object negates or stops to be an entity, it loses only those attributes that are strictly equal in value to the a priori data or typical attributes of that entity's object form. Otherwise, the object retains its unique attributes that are different from the a priori data of the entity.

3.3 Operations with Objects

Scenarios as entities, sets, or classes of processes are not formalizable as such, but a priori data about them is described in the form of entity objects $@Plot$. The structure of a scenario object differentiates a set of intention monads:
$@Intention$: $[(\#before_i@Variable,\ \#after_i@Variable)@Intention]$.

A variable as an entity differentiates inertia or state as such. That is, it does not point to an object, but to the state as such of objects outside of time, or in essence. When a scenario is realized or executed in time, variables are different into objects.

In this context, the differentiation (definition) of a variable can also be another variable monad, which allows for defining scenarios that modify other scenarios, thereby describing the evolution of the system.

The variables $before$ and $after$ are different as monads of the states as such of an object "in the past" and "in the future." Their difference, in turn, shows how the object changes over time.

Create Operation

An intention ((), $\#after@Variable$), which has no "before" state, is differentiated (defined) as an intention of creation or realization of a monad. The created monad is differentiated as an empty or unknown monad with a specified experience identifier. If no identifier is specified, one is generated uniquely upon the node's creation.

After creation, a modify operation is performed on the monad, where the $\#before@Variable$ state is an empty monad (), representing the unknown or unrealized state before the monad's existence.

Read Operation

An intention ($\#before@Variable,\ \#before@Variable$), which does not differentiate "before" and "after" states is differentiated (defined) as an intention of undifferentiated change. That is, reading the monad without changes during the realization or execution of the scenario.

Update Operation

An intention ($\#before@Variable,\ \#before@Variable$), which differentiates the "before" and "after" states, is differentiated (defined) as an intention of differentiated change.

The realization of the intention ($\#before@Variable,\ \#after@Variable$) in a monad $\#m$ is the monad: ($\#m$ / ($\#before\ -\ \#after$)), ($\#after$ / $\#before$))

That is, after the change, the monad $\#m$ is differentiated (defined) as the oneness of its difference by essence from the difference by relation of the "before" and "after" states, and the difference by essence of the "after" and "before" states.

In other words, the monad loses those attributes and entities that are different in the "before" state but not in the "after" state. It acquires those attributes and entities that are different in the "after" state but not in the "before" state.

Delete Operation

An intention $(\#before@Variable,\ ())$, which has no "after" state, is differentiated (defined) as an intention to delete the monad. The deleted monad is differentiated as an empty or unknown monad; that is, it is no longer differentiated (minded) and does not exist.

4. Technical Implementation

4.1 (Meta)ontological Interpreter

The formalisms of (meta)ontology are more abstract than machine language instructions or the matrix operators of a neural network. For implementation, an ontological approach was chosen, which involves describing at a meta-level an interpreter that works dynamically with various ontologies. These ontologies include not only entities but also their changes (scenarios, functions, predicates, etc.).

Since entities (and a priori data about them), scenarios, functions, reflexes (including neural network models), and objects are all represented by monads of the one structure, the implementation of the metaontology as a software complex must:

1. Compute the algebra of monads via an interpreter.
2. Ensure the realization of the scenarios, functions, and reflexes described in the ontology.
3. Provide an interface for users, as actor monads, within the framework of the ontologies.

Describing object states through attributes, as well as functions as mathematical operators or predicates, is trivial from an object-oriented programming perspective, for example. Further explanations will mainly concern the supra-rational phenomena associated with actors, scenarios, and entities, which by definition cannot be fully and unambiguously formalized. Thus, the implementation proposed below is knowingly incomplete and not the only one of its kind, even within the proposed formalisms of the (meta)ontology.

Intentionality, Actor, and Choice

In the next version of the document this section will contain the formalization of the actor as a monad and the description of the context of its choices as a finite automaton within the current state of the ontology.

Execution of Scenarios

In the next version of the document this section will contain the formalization of differentiating objects in variables and the execution of the monad algebra operations.

4.2 Realization of the Stated Tasks

Increasing Ontological Uncertainty

In the formalism of (meta)ontology, uncertainty is increased by creating empty or unknown monads. The subsequent elimination of this uncertainty, for instance in the form of hypotheses, involves their differentiation into specific scenarios, variables, objects, a priori knowledge about entities, etc., by means of a posteriori data or a priori judgments.

Proposal of A Priori Operations

Within the proposed epistemology of monads, a computational device is fundamentally capable of verification, or a posteriori data gathering through sensory interaction with the surrounding space, reflexive recognition of information, and rational calculations within the current ontology.

However, the device is incapable of a priori judgments due to its lack of consciousness and the intentionality that stems from it, which is necessary for the non-deterministic generation of hypotheses regarding the current state of the ontology.

Within the scope of device adaptation (maintaining its physical state within acceptable parameters) or tasks set by a human, it is possible to project a priori judgments into a priori data as hypotheses for subsequent a posteriori verification by the device itself, using operations from the algebra of monads.

Such operations are hereafter called **a priori operations**:

1. An operation of entities differentiation within objects based on current a priori data about those entities (recognition of entities objects within objects)
2. An operation of attributes changes differentiation within the entities objects based on the previous operation (recognition of attributes changes within entities objects)
3. An operation to compute a priori data about the entity based on a set of objects (objects' classification into entity objects).
4. An operation to compute a new scenario based on a desired state, given conditions, and the current state of the ontology (a search through the graph of scenarios and states).
5. An operation to compute a priori data about a new scenario from the difference of changes in object states over time (recognition of new variables and scenarios).

In the next version of the document this section will contain the formal description of a priori operations using the algebra of monads.

4.3 Architecture

The concept of a "cog" is differentiated (defined) as a local ontology or module. A cog realizes as one a set of monads as a project, a database of objects, a library of functions, interface components, and so on.

Currently, cogs are imported by copying their monads. In the future, the plan is to add an import mechanism via linking to between cogs in a read-only mode with version control. The emphasis in the import implementation is on minimizing dependencies between cogs at the expense of redundancy (in accordance with the principle of the locality of truth).

Functionally, the software complex differentiates the following components:

1. A database (base of monads).
2. A (meta)ontology interpreter or engine.
3. Models of reflexes (neural networks).
4. A client: interpreters for web and mobile interfaces defined as monads of components and scenarios within the ontology itself.

In the next version of the document this section will contain the Diagram of components.

Currently, a cloud solution with a web interface is available, providing a ready-made environment for creating and using cogs as separate projects. Several distributions of the software complex are planned:

1. Local distributions with various implementations of clients (web, native applications), reflexes, and databases for different tasks, with the possibility of horizontal scaling of functional components.
2. Optimized versions for embedded devices.
3. A cloud marketplace for cogs for their publication and monetization, available for import in the cloud and in local distributions.
4. A fully decentralized environment with a distributed cogs ledger provided by local distributions.

Further optimization in the form of specialized hardware and an OS is possible in the future.

5. Conclusion

The introduction outlined the problem of the limitations of determinism. Tasks were set for deterministic devices to bypass these limits by formalizing a non-deterministic increase in the device's state uncertainty and the generation of hypotheses not limited by its current state. Epistemological questions regarding the foundations and justifications for such formalisms were also raised.

As a solution, based on differential phenomenology, a formalization of (meta)ontology was proposed. This formalism allows, albeit in an incomplete and limited form, for operating with entities and scenarios at the same level of abstraction and as dynamically as with data.

Within the framework of this (meta)ontology, definitions for a priori and a posteriori data were proposed, along with criteria for truth and verification, the boundaries of rationality, and the fundamental limits of cognition for devices lacking consciousness.

An algebra of monads was proposed, describing changes in the states of objects (data), scenarios, and entities ontologically, without the use of programming code. This, in turn, allowed for the formalization of a priori operations, enabling the generation, however incomplete and limited, of hypotheses not determined by the current ontology.

Non-determinism in a deterministic device is not achieved within the device itself or through any formalism as such. Rather, **it is achieved** by the device using these formalisms **through interaction with non-deterministically changing being at a new level of abstraction** (mindfulness).

In doing so, it gains the limited ability to change its state and behavior in relation to the non-deterministic changes of being. Albeit deterministically (and therefore incompletely and sub-optimally), but nevertheless. For the purpose of its own adaptation (maintaining physical existence) and within the framework of realizing the intentions or tasks set by human consciousness.

6. Contacts

To discuss this paper and / or reach the authors you can join our

Telegram: https://t.me/CognitoProjectChat
Discord: https://discord.gg/rDMdWrYTWB
Or contact us by email at hello@cognito.one