mathematics

MDPI

Article

Representation and Abstraction

Antal Jakovác [1,2,*] and András Telcs [1,3]

[1] Department of Computational Sciences, HUN-REN Wigner Research Centre for Physics, H-1121 Budapest, Hungary; telcs.andras@wigner.hun-ren.hu
[2] Department of Statistics, Institute of Data Analytics and Information Systems, Corvinus University of Budapest, H-1093 Budapest, Hungary
[3] Economics, University of Pannonia, H-8200 Veszprém, Hungary
* Correspondence: jakovac.antal@wigner.hun-ren.hu

Abstract: In this paper we propose a mathematical model for cognitive phenomena such as abstraction, generalization and extension. The main concept is the coordination of the observable world with relevant features, a concept defined mathematically in the paper. These features represent necessary conditions for performing a task. There are different strategies to choose a coordination, and some of them can be associated with the System 1 and System 2 ways of thinking. System 2 uses multiple, context dependent relevant features for coordination, which makes it possible to speak about abstraction and the generalization of concepts. We also discuss the way extension works in these systems.

Keywords: relevant features; artificial intelligence; abstraction; generalization; extension; System 1 and 2 thinking

MSC: 68T30

1. Sorts of Intelligence

The process of AI development seems to arrive at a milestone. The traditional techniques resulted in marvelous achievements, both in generative and in classification models. Still, a human observer lacks some coherence, some structured thinking instead of an even very educated guess.

Indeed, it is worth recalling the modes of thinking introduced by D. Kahneman [1]. He argues that there is a fast, intuitive way of thinking (System 1), relying on heuristics and associations, which is used by the brain to rapidly assess situations and make judgments. However, it can lead to cognitive biases and errors due to its reliance on quick judgments rather than a deep analysis. In contrast, the System 2 mode involves slow, deliberate, conscious, analytic processing. It is used for complex problem-solving, critical thinking, and decision-making, which require careful evaluation of evidence and consideration of multiple factors. We use both of the modes: System 2 is used for planning and understanding the situation, while System 1 is helpful in instinctive operations and for pattern recognition tasks. The main focus of System 1 is to stay "in synch" with the external stimuli, while System 2 intends to give an adequate answer even if it comes somewhat later. If we compare these properties with the way of operation of present-day AIs, it is evident that we have built System 1 machines that are good in pattern recognition and instinctive decision-making but lack coherence and structured analytic understanding.

As the above cases also suggest, intelligence is not a homogeneous phenomenon. From another point of view, this is supported by the CHC theory of human intelligence [2], where the general intelligence (Stratum III) is subdivided into much smaller, individually

check for updates

Academic Editor: Francesco Calimeri

Received: 19 March 2025
Revised: 30 April 2025
Accepted: 6 May 2025
Published: 19 May 2025

Citation: Jakovác, A.; Telcs, A. Representation and Abstraction. *Mathematics* **2025**, *13*, 1666. https://doi.org/10.3390/math13101666

identifiable parts, with broad (Stratum II) and narrow (Stratum I) abilities. So, we may tell that there exist different sorts of intelligence, depending on the task we want to solve with it. Human intelligence is a very complex system of the different types of sub-type intelligence, as the CHC model suggests.

In the evolution of AI, machine intelligence was always created to answer specific questions and to solve some specific class of tasks. Alan Turing suggested that the best task to achieve human-like intelligence is if we could create a machine that talks like a human, at least according to a large enough part of human judges (the Turing test, [3]). Although there exist programs that successfully passed the Turing test [4], it is evident by now that this is not the singular task whose solution would lead to human intelligence.

Another class of tasks comes from the pattern recognition idea. The task is to find that element from a finite set (classes) that fits the best to the input. The definition of "best fitting" comes from a number of examples where the correct answer is provided. This idea works perfectly if the number of choices is limited, let it be so huge like the possible positions in the go or chess game, or the possible words (tokens) that would fit to the content of a longer text. As we have discussed above, the pattern recognition idea finally resulted in a number of System 1 machines, solving certain tasks even better than humans.

As it should be clear from the previous arguments, System 1 alone cannot approach human intelligence, where System 2 is an inevitable part of higher level thinking. We propose a framework that is based on the representation introduced in [5], aiming to reproduce some functions of System 2 thinking.

Neural networks have achieved excellent performance in many tasks, but it is becoming increasingly clear that even deep learning has some limitations (cf. [6]). Neural networks are universal approximators, performing dimension reduction and manifold learning. They essentially serve as exploration tools in a continuous universe but operate purely in discrete scenarios. The framework created by Jakovac [5] offers a complementary approach; its foundations are discrete, making it suitable for solving problems in discrete scenarios. Moreover, it can be applied in continuous scenarios thanks to the natural approximation of manifolds. In this paper, we utilize the capabilities of this framework. We demonstrate its practical applicability and the potential to perform abstraction, problem generalization, associative learning, and extension within it.

In the literature, there are several works that try to explore these topics in different contexts. The learning of proper representation in Large Language Models was studied in [7,8] and in a more general framework in [9]. Renormalization group ideas are used in [5,10].

The higher level cognitive processes are based on the proper representation of a limited part of the word (c.f. [5]), abstraction, generalization and extension, which are the building blocks of modeling and decision-making. In that respect, we follow Holte's [11]) approach. In all problem-solving tasks, we should have a well-defined universe Ω; $\mathcal{L}_U$ is the set of known universal laws, $\mathcal{L}_S$ contains the task specific laws, constraints, $\mathcal{F}$ is the set of all possible observations (features) of elements of Ω, and f is the operator mapping from space of the solution $\mathcal{S}$ functions to real values, basically the goal function of the task.

2. From Reality to Modeling

Let us first overview qualitatively the task of data representation and turn later to a mathematical formulation of the tasks. This consists of several levels of abstraction, from the lowest, observational level to the highest, which is a full model of the surrounding world. A schematic description of this process can be seen in Figure 1.

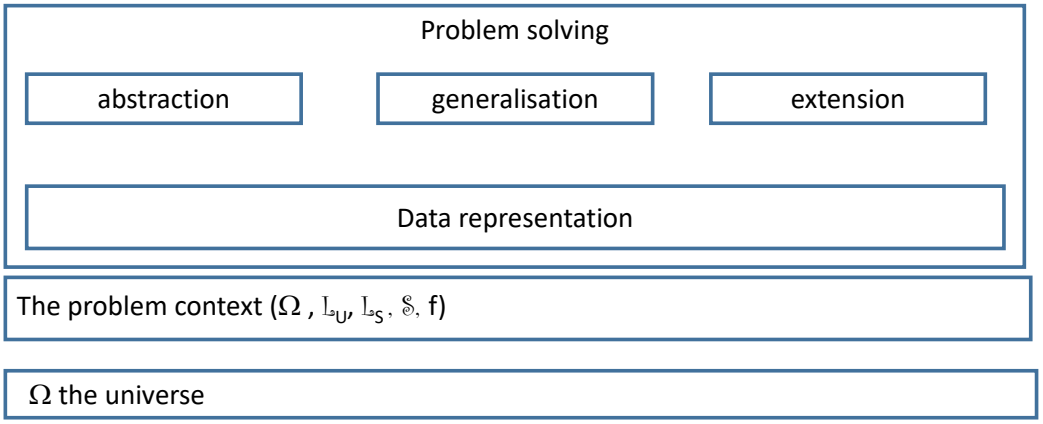

Figure 1. Schematic diagram of the problem-solving.

2.1. Reality

First, we need to make a philosophical disclaimer. In this paper, we will rely on an objective reality; thus, we assume that the world exists independently of any perception and other actions. We use an absolute time, which means that we single out a spacetime coordinate system, and use its local coordinates to split space and time (we do not deal with the problem of non-global spacetime mappings). Then, we can speak the "state of the world" as an initial condition on a Cauchy surface. We also do not deal with the problem of strict causality (i.e., we do not deal with finite signal propagation speed) and assume that an intelligent agent can react to the state of the world freely.

For a mathematical representation, we denote by Ω the set of the currently possible existing states of the world. Ω does not contain the dinosaurs, since they are not existing any more, and does not contain the unicorns since they never existed. The exact time evolution of the world is a $\Omega \to \Omega'$ transformation, but by including the possible future states into Ω (i.e., using $\Omega \cup \Omega'$ instead of both of them), the time evolution can be considered an $\Omega \to \Omega$ map. It should be emphasized that Ω also includes the state of the decision-makers. While Ω does not contain physical entities like dinosaurs and unicorns, they can still be present within it as concepts.

2.2. Native Coordination by Observations

The agents are informed about the world only through performing observations, and this is far from being complete information. In a living creature, the possible observations are preset by sensational capabilities; for humans, this includes seeing, hearing, touching, etc., all inputs from our seven senses. For an AI or ML system, the physical setup specifies the available information. The output of these observational "devices" produces raw inputs. We will consider sensational capabilities fixed in this work.

Mathematically, the observations of the universe lead to finite precision numerical results, which can be represented as a collection of zero-one coordinates $c_i : \Omega \to \{0, 1\}$. It may be worth mentioning that the collection of the coordinates $C = \{c_i\}_{i \in \{1,\ldots,N\}} : \Omega \to \{0, 1\}^N$ is neither injection, nor surjection: different states may lead to the same observations (e.g., states that differ in quantum-state details), and there are coordinate configurations that do not code anything in real life. Nevertheless, in the sequel, we will ignore this subtle difference and identify Ω and $\{0, 1\}^N$.

2.3. Context

The universal goal of any living creature is to maintain and reproduce itself. The environment produces challenges for the the system (human, animal, virus population, AI

entity, etc.). The combination of the environmental conditions and the aforementioned universal goal defines tasks and problems to solve.

An intelligent agent solves a task by reacting somehow to the environment and changing it. Its behavior, action, thus can be represented by an

$$f : \Omega \to \Omega. \tag{1}$$

For the sake of simplicity, we factorize and simplify such action, using the ideas of the Markov Decision Process (MDP c.f. [12]). We choose basis action functions f_j^{base} and decide whether we perform that action or not. The decision is a binary function $g_j : \Omega \to \{0, 1\}$, and

$$f_j = g_j f_j^{base} + (1 - g_j)\mathrm{id}. \tag{2}$$

That means that if $g_j = 1$, then we perform the $\tilde{f}_j$ action, otherwise not. Then, the complete action is represented by

$$f = f_k \circ \cdots \circ f_2 \circ f_1 \tag{3}$$

In this decomposition, we assume that f_j^{base}s are "given", defined by the context, and the only "task" is to make the decision g. In that way, the action itself can be very complicated (coded in f_j^{base}-s), but the solution of the problem is a sequence of binary decisions (task processing, of course, can be parallelized, but here we use the flatted description, which clearly can represent other, more complex processing structures).

It should be noted here that Ω contains the state of the problem solver as well and has memory. In that respect, we slightly deviate from the standard setup. Here, the action (the result of the action) is not separated from the actor, from the actor internal state, but considered as a part of the universe. In that way, reinforcement, feedback, top-down readjustment, attention shift and transfer learning are inherent in the decision and task problem-solving process.

Another remark is that while we are speaking about MDP, it does not mean that the steps are stochastic. Some of them can be deterministic (that is also a Markov step if it depends only on the actual state), while others can be random (a higher order Markov process always can be converted to a first order one on the price of increasing the dimensionality). A decision with a random component can arise, for example, in the case of Bayesian modeling or in the case of a mixed strategy.

In order to keep our description as simple as possible, the reward and expected discounted future reward are incorporated into the single goal function. The calculation of the actual expected discounted reward can be very complex, and the decision based on that can be similar to stochastic or dynamic programming tasks. Humans excel in solving such tasks, but here we do not elaborate on that branch of decision-making (see again [12]).

Let us make another disclaimer here. The above setup is a finite approximation of possible infinite precision values of the observations. The universe is also limited to a finite number of objects and a finite number of their observable features. We think that these restrictions do not pose serious limitations, in practice, in real world scenarios in describing cognitive processes and decisions. If, for a given task, a decision, an action, is a good approximation (with respect to the problem goal), the action can be successful regardless that the solution is not exactly the optimal one.

To mention some examples, for a predator, a hunt consists of base actions like "sneak", "stalk", "wait", "run", "chase", "turn" and "stop". All actions are complicated, but the predator decides whether, for example, to start the chase (1), or not (0), which is a binary decision. Here, we can immediately see that the decision depends not only on the environ-

ment but also on its internal state and memory. If the chase consumes too much energy, the predator runs out of oxygen or gets overheated, it will stop when reaching a (built-in or learned) threshold. The predator's memory also influences the decision second by second. By recalling past successes or failures, and considering factors such as the current distance, the prey's fitness, its own fitness, a Bayesian estimate of success probability, contributes to the decision (c.f. [13]). For the prey, the hunt means a different context since it happens unexpectedly, and it must be cautious. For the prey, the possible actions are "graze", "listen", "look", "flee", etc.

To have an AI example, classification into two classes is a task, where the action is $f(0) = \{\text{sort into the first class}\}$ and $f(1) = \{\text{sort into the second class}\}$.

2.4. Features

The goal of intelligence is to perform the best possible action, where "best" means optimal from the point of view of the task, or reward. This means that it should estimate the optimal f action function (or G goal function) in the best, or at least in a good enough way.

By representing the decision mechanism by an MDP, the performing of a base action depends on the value of the corresponding decision funciton g. Since the range of g is $\{0,1\}$, it naturally divides Ω into two disjoint parts $\Omega = g^{-1}(0) \cup g^{-1}(1)$. In practice, it is more frequent not to perform an action than that to perform it, so $|g^{-1}(0)| > |g^{-1}(1)|$.

It seems to be useful to coordinate Ω in a way compatible with the above division. This leads to the definition of a relevant feature:

Definition 1. *An $a : \Omega \to \{0,1\}$ function is a relavant feature of a quantity $g : \Omega \to \{0,1\}$, if $a^{-1}(1) \supset g^{-1}(1)$*

Formally a relevant feature is a coordinate of Ω (c.f. the definition of c_i above). Relevant features may play a lot of roles; below, we listed some of them.

- Necessary condition: from the definition, if for an $\omega \in \Omega$ we have $g(\omega) = 1$, then necessarily we have $a(\omega) = 1$. Thus, $a = 1$ is a necessary condition for $g = 1$.
- Conserved quantity or law: for all $\omega \in g^{-1}(1)$, we have $c(\omega) = 1$. Thus, a is a constant quantity over $g^{-1}(1)$, or $a(\omega) - 1 = 0$ is a law satisfied over $g^{-1}(1)$ set.
- Selective coordinate: to represent g, we shall collect as many (M) relevant a_i coordinates that $\cap_i a_i^{-1}(1) = g^{-1}(1)$. Then, to decide whether $\omega \in g^{-1}(1)$ or not, we have to check that $a_i(\omega) = 1$ for all $i = 1 \dots M$.
- Abstraction: by finding a relevant feature, we introduce a *concept* that did not exist earlier for the representation of a phenomenon. Therefore, it can be considered as the prototype of any abstractions.

Those functions for that $a^{-1}(1) \cap g^{-1}(1) \neq \emptyset$ are not useful to represent g, so they are called *irrelevant* features. Since irrelevant features split $g^{-1}(1)$ into two parts, they can be useful if we want to tell apart the individual elements of $g^{-1}(1)$; therefore, they can be called *descriptive* features as well. For example, when we want to compress images belonging to a given class (e.g., natural images), relevant coordinates are constant, so we shall only remember the irrelevant, descriptive coordinates.

In papers [14,15], it is proven that if we found the maximal number of statistically independent relevant coordinates, then the rest of the coordinates are irrelevant. We can also assign an entropy to the representation using this theorem.

To give a simple example, to decide whether an animal is a cat, the presence of the spine is a necessary condition. Mathematically, it means that $g^{-1}(1) = \{\text{cats}\}$ is a subset of $a^{-1}(1) = \{\text{has spine}\}$.

In the whole sequel, we will use the terms "relevant" and "selective" interchangeably, as well as 'irrelevant' or "descriptive". In the given context, we can omit the irrelevant features, which leads to an information reduction. This information reduction, on the other hand, is the basis of the abstraction and generalization, which we will discuss later in more detail.

3. Search for a Representation

The choice of the relevant features is not unique, and there can be different strategies that can be optimal from different points of view.

3.1. System I Mode Features

The simplest choice for choosing a relevant feature for a g decision function is $a = g$. For example, if we want to recognize cat and dog images, we may use two coordinates, a_{cat} and a_{dog}, which are one exactly when we see a cat or a dog in the image, respectively.

In human thinking, the System I mode uses this type of approach, and it can be very effective in certain situations. Since this solution has the power of having the lowest number of relevant variables, its evaluation is very fast and simple. It can be also very precise for performing specific tasks.

The present-day AI solutions, as it was discussed in the Introductions, also work in this way. In practical applications, we do not have strict zero and one values but a $p \in [0, 1]$ real number, which can be interpreted as a probability of belonging to a given class. The actual class is the one with the largest p.

There are, however, numerous disadvantages of this choice. Staying at the example of cat and dog image recognition, we encounter the problems listed below:

- In real world $|g_{cat}^{-1}(1)| \ll |g_{cat}^{-1}(0)|$, i.e., there are vastly fewer cat images than non-cat images. Neural networks work with continuous (even differentiable) functions in the sense that if its value exceeds 0.5, then $a_{cat} = 1$, and vice versa. However, the task to define an analytic function that is smaller than 0.5 *almost everywhere*, but the important information is the domain where it is larger than 0.5, is not solvable. We can reasonably train a continuous function that is smaller and larger than a given value on domains with comparable sizes. In the language of Machine Learning, training needs balanced datasets. That is why this logic is not suitable to separate cats and no-cats.
- If we just want to tell apart dogs and cats by comparing their respective probabilities, the function that realizes this task can be very complicated. This means that they are very tough to find (long training times).
- Another consequence is that the purpose of a_{cat} and a_{dog} is to tell apart dogs and cats. Therefore, they are very task-specific. This means that, even for a similar task, we shall use a different coordination (catastrophic forgetting).
- Since cats and dogs are separated by a single real number, the decision can not be self-verified, and so approximation failures remain undetected (error prone, c.f. unpredictable failures in AI classification).

Because of these disadvantages, one cannot base the complete mental system only on System I features. It is used in nature for tasks important enough to develop their own features that are steady in time and that require a decision between nearly equally probable cases.

To give an example, face recognition is important enough, and it is steadily used; additionally, within our environment, we meet with the same people. Therefore, humans developed System I features for face recognition. However, this system can fail when recognizing faces from another race (e.g., Europeans may have difficulty recognizing Asian faces, and vice versa) or if we have not seen a person for a long time.

3.2. System II Features

There are other ways to choose relevant features; in the rest of this paper, we will deal these techniques. The main idea is to find simpler, less precise, but more robust, features that can be useful not just in a given task but also for other, similar tasks.

To make it clear why it is needed, let us consider two simple examples. Imagine returning home in the late afternoon. We might or might not turn on the light in the living room. This decision falls into the System 1 category, almost automatic, based on our habits and comfort. Another example comes from psychology: Alice and Bob are planning a date. Alice suggests going to the cinema, while Bob proposes a basketball match. It is evident that this decision requires more than a quick, instinctive choice. Considering the appeal of their own suggestions and the potential disappointment, one of them cancels the date, feeling upset. Unlike the automatic decision about the light, this situation demands a System 2 decision.

To establish a proper decision for an action, we need a much richer analysis of the situation. For the proper decision, Bob needs his earlier experience of Alice's flexibility, his own alternatives to visit another match at another times and conflict resolution schemes he acquired in different situations, e.g., proposing a third alternative that might be satisfactory for both of them. Some schemes are applicable, while others fit only, say, to business situations. That way of problem-solving needs an abstraction of the problem (it is not interesting what is Alice's or Bob's actual suggestion, only the difference, only the conflict counts). Needs generalization, there might be more alternatives, more conflicting parties and maybe even extension, is needed incorporating more alternatives or scenarios.

3.3. Representation

A typical problem is confined to an unknown subset Ω_1 of the universe Ω, and the solution needs the correct representation of it. We may take the classification problem for $\{\Omega_1, \bar{\Omega}_1\}$ itself, or it may be the first step of a more complicated problem-solving. Let us assume that $|\Omega| = 2^N$, and $\Omega_1 \subset \Omega, |\Omega_1| = 2^{N_1}$. The understanding of Ω_1 means that we have an $a : \Omega \to \{0, 1\}$ canonical representation, a rearrangement of the coordinates $c : \Omega \to \{0, 1\}$. Following the notations and discussion of [15], we call a representation canonical if it contains all the independent selective and descriptive coordinates, and the descriptives are uniformly distributed over Ω_1. For convenience, we assume that in the canonical representation, the coordinate sequence starts with the selectives:

$$\overbrace{a_1, \ldots a_M}^{\text{selectives}}, \overbrace{a_{M+1}, \ldots a_N}^{\text{descriptives}}.$$

A training sample A_K ($|A_K| = K$) is error-free if all elements are from Ω_1 (i.e., $A_K \subset \Omega_1$), so the sample is not polluted by elements of $\bar{\Omega}_1$ (the complement of Ω_1). If $K = 2^{N-N_1}$, we are finished. We show that a much smaller sample suffices if the sample is error-free. Let i be a coordinate in which all the sample elements are constant. We estimate the probability that it is not a selective coordinate.

$$\begin{aligned} p_b &= P(c_i = \text{const in } A_K | i \text{ is descriptive}) \\ &= 2^{-K} \end{aligned}$$

We know that N_1 coordinates are surely constant; the only problem that can arise is if some extra coordinates are erroneously constant. That means that if we found constants in $L > N_1$ coordinates, then $L - N_1$ are bad.

$$P_S = P(\text{all the constant coordinates are selective} | A_K) = (1 - p_b)^{L-N_1}$$

Elementary calculation shows that for

$$K > -log_2(1 - ((1/2))^{(1/(N-N_1))})$$

$P_S > 1/2$ and a larger K-s ensure the correct identification of the selective coordinates with high probability.

This is a not very surprising but promising conclusion. It is expected, given that the probability, p_b, finding a descriptive coordinate constant in our sample, is exponentially decreasing with K. At the same time, it is a promising result since it shows that it is not necessary to see all of the elements of the set that we would like to represent (understand) but a relatively small sample will suffice. Let us note that the proper K depends and increases with $N - N_1$, but the increase is slow.

3.4. Mixed Sample

Assume that the sample contains elements from $\Omega \backslash \Omega_1$ as well, and we do not necessarily know the cardinality of Ω_1 and Ω, but we use N "features", coordinates for their description. We also assume that $|\Omega_1| \gg |\Omega \backslash \Omega_1|$ and the selective coordinates for Ω_1 are equal to one (but we do not know which are those coordinates). The sample of n elements contains $n \times p$ elements from $\Omega \backslash \Omega_1$. We develop a Baysian decison for the best representation. Let k_i be the number of ones at the i-th coordinate. Let us rearrange the coordinates for the sake of convenience in a descending order of k_i. In the new coordination, then $k_1 \geq k_2 \geq, \ldots, \geq k_n$. We seek for the optimal choice of the selective coordinates. We claim that it is better to chose i over j if $k_i \geq k_j$. Following that, we try to determine an index $M \geq 1$ so that the first M coordinates (in the reindexed order) are the selective ones:

$$\overbrace{a_1, \ldots a_M}^{\text{selectives}}, \overbrace{a_{M+1}, \ldots a_N}^{\text{descriptives}}.$$

We claim that if we are seeking for a representation with M-selective coordinates, this is the best one (we neglect the little ambiguity if around c_M we have other coordinates with the same value).

We have elements from Ω_1 and from the complement with probability $1 - p, p$ respectively. Here, we have to make an important assumption. All the samples are chosen uniformly from Ω_1 and from the complement, and the choices are also independent. If the choice is not uniform over Ω_1 or $\Omega \backslash \Omega_1$, the sample might over-represent a subset and result in a false conclusion. We know that all the elements from Ω_1 have exactly $N - N_1$ coordinates in which the representation is one, so we have a $G \sim (1 - p)n$ sample item from Ω_1, and, at the selective coordinates, there are a $G \leq |\Omega_1|$ number of ones. Of course, an indifferent (descriptive) coordinate contains n ones with binomial distribution $B(0.5, n)$. Of course, if $p = 0.5$, a selective coordinate will mimic the behavior of a descriptive one, but we think that it is reasonable to assume that $p << 0.5$. If that is the case, elementary statistics help to separate the selective and descriptive coordinates.

If i is a selective coordinate, then $a = a_i$ is approximately normally distributed with mean $n(1 - p)$ with a standard deviation $\sqrt{np(1 - p)}$, while if it is a descriptive coordinate, it is approximately normally distributed with a mean $0.5n$ with standard deviation $\frac{1}{2}\sqrt{n}$. The two cases can be separated easily with a t-test. If p is relatively small and n is large enough, M can be identified with high confidence.

Let us list some cases in order of increasing complexity.

1. If $p = 0$, then in the selective coordinates, $a_i = 0$-s are just the number of samples $n \leq |\Omega_1|$, while in the descriptive coordinates, we have binomially distributed a-s with with sample size, and $q = \frac{1}{2}$ chance of success ($B\left(\frac{1}{2}, n\right)$. So, $M = i$ is the last index for which $a_i = n$.
2. If $p << 0.5$ and n is large, then the selective coordinates we have are $a_i \in B(1-p, n)\ \tilde{}N\left((1-pn), \sqrt{np(1-p)}\right)$, while for the desciptives, we have $a_i \in B\left(\frac{1}{2}, n\right)\tilde{}N\left(\frac{1}{2}n, \frac{1}{2}\sqrt{n}\right)$. In this case, M can be selected again as the last index where the a_i-s are close to $(1-p)n$.
3. Case two needs the study of the size of the sample. Let z_α be the usual upper tail cut-off value for the tail probability α. Then, if $(1-p)n + z_\alpha\sqrt{np(1-p)} < |\Omega_1|$, the probability that we have all the elements of Ω_1 in the sample is less than α. If $(1-p)n - z_\alpha\sqrt{np(1-p)} \geq |\Omega_1|$, we have a chance to see all the elements of Ω_1, resulting in $a_i = n$ in the selective coordinates, which still clearly can be separated from the descriptive coordinates $a_i\tilde{}N\left(\frac{1}{2}n, \frac{1}{2}\sqrt{n}\right)$.
4. If $p << 0.5$ but n is small, the confidence intervals of the two distributions $N\left((1-pn), \sqrt{np(1-p)}\right), N\left(\frac{1}{2}n, \frac{1}{2}\sqrt{n}\right)$ may overlap, and the demarcation between the selective and descriptive coordinates is blurred. Straightforward calculation shows that that can happen if

$$n \leq \frac{z_\alpha\left(1 - 2\sqrt{p(1-p)}\right)}{1-2p}\tilde{}\frac{c}{1-2p}.$$

That is not a serious problem until p is well separated from 0.5, but that case can be clearly excluded; otherwise, the shown sample is not really representing Ω_1.

As a conclusion, we can say that in typical situations, the selective and descriptive coordinates can be clearly distinguished by the hitting rates.

3.5. Invariance and Representation

In what follows, we will refer to a bijection $b : \Omega \to \Omega$ as a transformation.

Definition 2. *We say that a coordinate $a = a_i$ is invariant under the transformation b if for all $\omega \in \Omega$ $a_i(\omega) = a_i(b\omega)$.*

Definition 3. *We say that a transformation b represents the set Ω_1 if the selective and only the selective coordinates of Ω_1 are invariant for b.*

It is immediate that b represents Ω_1 if and only if $b\Omega_1 = \Omega_1$.

Definition 4. *Let b^k denote the k times iterated application of the transformation b.*
The orbit of b is the set $O_b(\omega) = \{b^j\omega$ for $j \in \mathbb{N}\}$ for a given $\omega \in \Omega$.

It is clear that if $O_b(\omega) = \Omega_1$, then b defines a permutation on the elements. This permutation is given by the rearrangement $b(\Omega_1)$ of Ω_1. In that way, we have a one-to-one correspondence between coordinate representations and representing transformations. Let us note that if b represents Ω_1, then it represents its complement as well.

Again, it is clear that if there is an $\omega \in \Omega_1$ s.t $\{b^j\omega$ for $j \in \mathbb{N}\} = \Omega_1$, then $b\Omega_1 = \Omega_1$. Consequently, such covering of Ω_1 with the iterates of b ensures that b is a representation of Ω_1.

As we have discussed in Section 3.3, a relatively small sample from Ω_1 provides strong evidence for the selective coordinates. If b represents Ω_1, a small sample can be enriched using b. Of course, the gained information depends very much on how b is mixing. Good mixing means that b is covering Ω_1 uniformly in relatively small number of steps. The worst scenario is if Ω_1 partitions into b-invariant subsets; in that case, the full covering from particular starting elements is not ensured.

Definition 5. *We say that b is uniformly mixing if, for almost all $\omega \in \Omega$, then $b\omega$ is uniformly distributed in Ω.*

Let us observe that a representation with coordinates is one kind of "knowledge" about a set; a transformation is an other one. With the aid of a representation, we can decide for any $\omega \in \Omega$ if it belongs to Ω_1; while using b starting from one element of Ω_1, we can recursively find more and more elements of Ω_1, and we "learn" Ω_1. As it is explained in Section 3.3, one new element "learned" using b contributes to the exponential decay of the wrong representation probability.

In practice, it might be unclear if b is representing Ω_1. How can we verify it? Let us assume that b is representing Ω_b, and b is uniformly mixing. Let us denote by S_b the set of selective coordinates of S_b that we can obtain with high confidence by the iteration of b. Also, let S_1 be the set of the selective coordinates of Ω_1. Let $D = \Omega_1 \bigcap \Omega_b$. If $\frac{|\Omega_1 \triangle \Omega_b|}{|D|} \approx 0$, then Ω_1 and Ω_b are very similar, and we may accept b as a good representation of Ω_1 and S_b as the set of selective coordinates of Ω_1. So, we investigate the case when $\frac{|\Omega_1 \triangle \Omega_b|}{|D|} \not\approx 0$. First, let us assume that $\Omega_1 \not\subset \Omega_b$. This situation can be detected only if we have at least one $\omega_c \in \Omega_1 \setminus \Omega_b$. Big enough uniform sampling can ensure the existence of such an element. Let us emphasize that we should know that those elements are from Ω_1. Of course, here, we have a trade-off. It can be that the sample that ensures an element $\omega_c \in \Omega_1 \setminus \Omega_b$ is big enough to build S_1 with high confidence. In this situation, there is no essential role of b, but still it can improve our confidence in the validity of S_1. On the other hand, if such a sample is still small to build S_1, we utilize b as follows. With the aid of b, we can have S_1, but if we start an iteration from ω_c, we shall find inconsistency with S_b and reveal that b is not a good representation of Ω_1.

If $\Omega_1 \subset \Omega_b$, let us revert our consideration and study $\overline{\Omega}_b \subset \overline{\Omega}_1$, and we can use the same argument to check if b represents $\overline{\Omega}_1$, which at the same time means that b represents Ω_1 as well.

We see how important are the transformations and how useful it is if we find a uniformly mixing one.

4. Abstraction

Abstraction is one of the key tools of problem-solving, and it is a subject of extensive study in the artificial intelligence literature [16]. In general, abstraction is based on a ground universe and a target universe, and the goal is to solve a problem in the ground universe by lifting up the problem using abstraction to the target universe, solve the corresponding problem there and transfer the solution back to the ground universe, where the solution is still valid. In what follows, we give one possible abstract definition of abstraction. Formally, it seems to be a state abstraction [17], but, in fact, our universe and coordinate system can encode relations between objects, structure [18], laws and parameters to be the selected solution constraints. Equipped with such attributes, our definition covers domain abstraction, problem domains and semantic abstraction. Only the goal cannot be incorporated into our universe that is inherently external. That is the context, the problem we face, and it determines our observations and the definition of the ground universe with all the above-mentioned attributes.

Now, we give the abstract definition of abstraction. We assume that Ω is a big countable finite or infinite set, and its full description can be done by $f : \Omega \to \{0,1\}^C$ (where $C = \{1, \dots N\}$ or $\mathbb{N}$) index set. In that universe, we may have finite and infinite subsets. Clearly, finite subsets can be represented as before, while it is impossible to do so for infinite sets. We would like to represent Ω_1, a very big or infinite subset. Infinite sets may or may not have a finite set of selective coordinates. Of course, we cannot describe or record all (very long or infinite) coordinate sequences. Similarly, we cannot repeat a transformation infinitely many times to obtain proper representation. Still, we have a possibility. The selective coordinates can be approximated in the possession of a sample, similar as we have carried out in the finite case. The selective coordinates are the common pattern that we find in the sample (given to us or created by a transformation), and we create the concept of all the objects that share that pattern, the selective coordinates with the same value. If the sample is large enough, we may tend to say that the constant coordinates $C_1 = \{c_1, \dots c_M\}$ describe well the set. C_1 is the abstraction of the elements of Ω_1. It is not a set in Ω but a subset of coordinates used over Ω. We expect that all the objects that "belong" to the concept C_1 share all the aspects that we consider important, invariant against any transformation that is "relevant" with respect to the concept C_1. Of course, increasing C_1 further provides more and more abstract notions that contain Ω_1 and can be useful in problem-solving.

We remark that the selective coordinates are not unique. Indeed, any function $c' : C \to \{0,1\}$ where $c'(C_1) \mapsto \{1\}$ can be considered a new selective coordinate that can substitute any of the elements of C_1. Finding those selective coordinates (abstractions) that can be useful in a wider context is the task of generalization; see next section.

This definition is provided within the framework of understanding through coordination, and it is fully aligned with the "formal" accounts of abstraction as presented by Wright (see Hale [19]).

Now, let us assume that we have a function, denoted as f, and an equivalence relation, represented by R:

$$f(a) = f(b) \textit{ if and only if } R(a,b).$$

In this context, equivalence classes or invariant classes serve as abstractions of their elements. It is important to highlight the concept of invariance. In many instances, invariance represents the observations we make, which are subsequently followed by the creation of names and concepts that represent these invariant classes. In order to find a good abstraction, invariance transformations can be very useful. The repeated application of the transformation may show the contours of the invariant classes and provide a hint for coordinates to keep and coordinates to drop.

Finally, let us give an example of a universe in which relations between objects are encoded and a proper abstraction might be also richer than a state abstraction. Consider the set of pairs of people and label them by $0, 1, 2, 3$. Let 1 indicate that the pair has common parents, 2 indicates that one parent is common and 3 that their parents are not common, but the parents live together. The edges labeled with $0, 1, 2, 3$ define the abstraction of non-siblings, siblings (own siblings), half-blood (half) siblings and step siblings. This is a typical example when the domain has structure, represented by a graph, with labeled edges.

5. Generalization

5.1. Domain Generalizations

In our formalism, we have the next form of domain generalization. For the sake of simplicity, we assume again that Ω is finite. As we have seen, if the sample is restricted to a subset of our target, we inevitably create a bad representation. For example, we identify more selective coordinates due to the biased sample representing only a part of our target.

The opposite direction is more interesting when a given representation S_1 of Ω_1 fails to cover some important species. Imagine that we have $\Omega_j, j = 1, \ldots, J$ subsets and some representatives from each of them. We are looking for the joint generalization that covers all of those species. Here, we seek those selective coordinates that are common across all the new species. The new set of selective coordinates S^* defines Ω^* in a natural way:

$$S^* = \bigcap_{i=1}^{J} S_i$$

$$\Omega^* = \{\omega : c_k(\omega) = 1 \textit{ for all } k \in S^*\}$$

Here comes the key link between abstraction and generalization. The set of coordinates S^* is the joint abstraction of the seen elements and Ω^*. In this formal definition, domain generalization seems an easy task. The problem stems from the choice of the features, which may depend very much on the $\Omega_j, j = 1, \ldots, J$ subsets. It can be that the individual sets are characterized very well by different features, but the common set is too narrow due to the slight differences in how the basically same properties are captured by the features.

Domain generalization is present in our daily practice. However, it poses some risk. We see black cats, dogs and cows and conclude that "all animals are black" and that the set of animals will be the set of black ones. That is called (naive) induction. There is no question that such intuitive induction can be very useful (quick and straightforward), but far from reliable. As we shown in Section 3, if the sample size is reasonable (for each subset), such mistake can be avoided.

Sometimes it is not easy to distinguish abstraction and generalization, particularly if we are actually in the problem-solving process. After all, domain generalization includes finding correct common abstract concepts. It is more important to observe that a concept of an abstraction contains less information than the items it refers to, while in the generalization, there is no information loss.

5.2. Task Generalization

In what follows, we try to grab another form of generalization, the process that helps to solve a task. If the task is to characterize (classify) unseen elements, then we face a domain generalization. If the task contains laws, structural properties or boundary conditions, then the generalization is more complex. For the sake of simplicity, we assume that the task is based on one subset Ω_1 and does not involve simultaneous solution on different domains. The generalization to the latter scenario is formally straightforward but can be very difficult in practice.

First of all, let us introduce the notion of laws. Laws over Ω are formulated using functions (operators), relations, bracketing and logical symbols. Let us skip the details. In short, we have a language $\mathcal{L}$, and we have laws described in that language as function: $f(\nu)$, here, $\nu \in (\Omega_1)^d$ for a $d \in \mathbb{N}$ and $f(\nu) = 0$ means that ν satisfies that law (the statement represented by the function is true). Such laws can be true for all elements of the ground universe Ω, or they may be characteristic to a given Ω_1.

From now on, we deal with the triplet $(\Omega, \mathcal{L}, C)$, where the coordinates (or features) have extended meaning and can be grouped according to their role.

$$C = C_R \cup C_L \cup C_T$$

where C_R contains the selective, representative coordinates of the subset Ω_1. The set C_L contains the coordinates encoding laws valid on the whole Ω, and C_T is the set of the task specific set of laws, coordinates. It contains structural properties laws (relations etc.)

specific to the subset (constraints, boundary conditions, etc.) specific to Ω_1 and the task. Let us emphasize here that the features can be as simple as color or size but can be as complex as mathematical or logical statements or biological or physical laws. The latter one has got much attention in the development of the physics-informed AI literature (c.f. [20]).

The task-specific generalization is far from being straightforward. It is not necessarily a loopless process. It consists of the following steps.

1. Representation of the base set (obtaining C_R);
2. Abstraction of the base set by removing some elements of C_R and work with $C_A \subset C_R$;
3. Creation of task abstraction C^* by removing some elements of C_A and C_T but usually keeping C_L;
4. Search for $\tilde{C}$ task abstractions similar to C^*;
5. Combine $\tilde{C}$ and C^* to obtain $\hat{C}$;
6. Find $\hat{\Omega}$ corresponding to the selective coordinates $\hat{\Omega}_R$ and the solution of problem based on $\hat{C}$;
7. Transfer back the solution to Ω^* (if failed, return to 5. or 4. or terminate);
8. Apply solution in the ground universe to Ω_1 (if failed, return to 5. or 4. or terminate);
9. Fix and store the found task abstraction $\hat{C}$ and the solution for later usage.

In that process, not only the selective coordinates can be sorted out, but some of the task-specific laws (and constraints) too. Moreover, at step 5, some further laws might be incorporated.

As a demonstration of the task generalization process, we present four very elementary IQ-test-like tasks. The task in each is to continue the series of simple objects as Figure 2 shows.

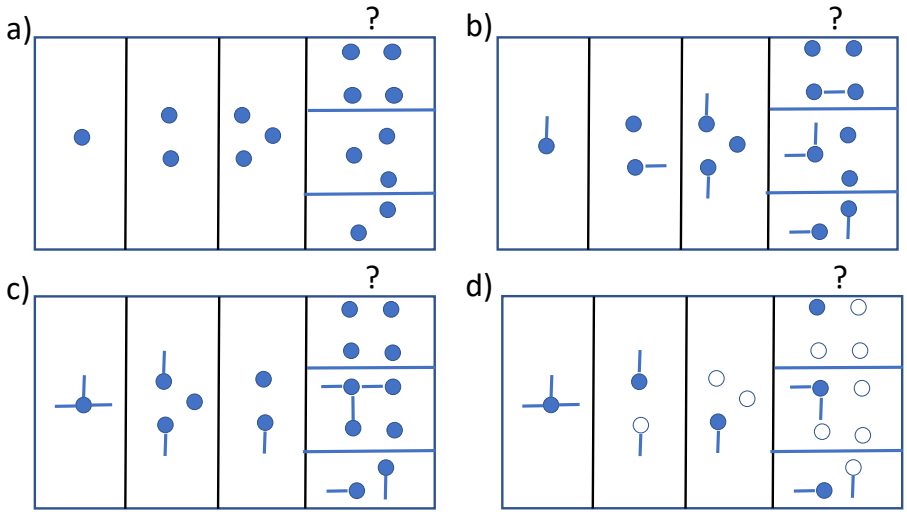

Figure 2. Series continuation task as an example. The four tasks follow four different rules, which are explained in the text. The key to solving the task is to recognize whether the observed features (such as the number of blue and white dots, or the number and position of the needles) are relevant or irrelevant in establishing valid rules.

In Figure 2a, the simple task is to continue the counting of dots. In Figure 2b, the same task is perturbed with the added little needles; here, the generalization is in the omission of the coordinates capturing the needles. Similarly, in Figure 2c, the count down of the needles is the rule, and the number of dots is the feature to be omitted. Finally, the solution of task in Figure 2d needs the observation that the count of empty dots and the total number of dots leads to the same conclusion, but we cannot distinguish two possible choices. The abstraction in this case can be the drop of one of the features or the usage of their joint. If we use another law learned in task (c), in the count down of needles, we find

the right solution. So, in the last task, not only a new kind of abstraction is needed but a combination of two previously seen laws are the key to the solution.

We shall emphasize that these types of tasks are solved using "simple" rules. Of course, we can create an ad hoc rule in favor of any choice, but we do not feel it is robust enough to accept as a good solution. This heuristic comes from real life, where, from limited data, we are only allowed to infer simple laws. If there is no such simple rule, we shall say that the problem has no solution rather than argue in favor of a more complicated one.

Let us demonstrate how the algorithm described above works in this logic problem. First, we characterize the base set. In example (a), we see dots, so the characterization of the base set is dot, dot–dot, dot–dot–dot, respectively. It already contains some abstraction since the relative positions of the dots are not listed. The next level of abstraction is that we introduce the number of dots, and we abstract it as $1, 2, 3$. Then comes the task that we should continue this series according to some rule(s). So, we need to filter out some laws from the examples given. It is clear in this case: $c_{L,i} = d_{i+1} - d_i - 1 = 0$ for $i = 1, 2$, where d_i is the number of the dots in the i-th slot. We generalize this rule to any i. This means that we assume $c_{L,3} = 0$, which results in $d_4 = 4$. This is the solution of the task in the abstract space. We now pull back the solution to the original space; this means then dot–dot–dot-dot in some arrangement. And, indeed, we find that there is an image corresponding to this description. Mission completed.

We can also establish a pseudo code for the process:

- Task: select the correct image.
- Base space description ("d" means dot): d, d–d and d–d–d.
- Abstraction: 1, 2 and 3.
- Law: $c_{L,i} = d_{i+1} - d_i - 1 = 0$ for the observed $i = 1, 2$.
- Law generalization: $c_{L,4} = 0$, yielding $d_4 = 4$.
- Pull back to the original space: d–d–d–d.
- Validity check: only one image corresponds to this description ✓.
- Task solved.

In a similar manner, we can approach case (b). We may suspect that the direction of the needles plays a role, but first we omit this possibility and write for the core part:

- Base space description ('d' means dot, and "n"means needle): dn, d–dn and dn–d–dn.
- In the abstraction, it is not clear which coordinates are relevant. We can use the number of the constituents: (1,1), (2,1) and (3,2).
- We find just one easy law: $c_{L,i} = d_{i+1} - d_i - 1 = 0$, where d_i is the number of the dots. We do not find simple rule for the number of the needles.
- Law generalization: $c_{L,4} = 0$, yielding $d_4 = 4$, and the coordination (4,?).
- Pull back to the original space: d?–d?–d?–d?.
- Validity check: only one image corresponds to this description ✓.

In this case, it turned out that the number of needles is an irrelevant coordinate, although first it seemed to be relevant.

Let us see case (c). We suspect that the direction of the needles does not play a role, so we try the following:

- Base space description ("d" means dot, and "n" means needle): dnnn, dn–d–dn and d–dn.
- In the abstraction, it is again not clear which coordinates are relevant. We use the number of the constituents as before: (1,3), (3,2) and (2,1).
- Law: we now are able to find an easy law for the number of needles $c_{L,i} = n_{i+1} - n_i + 1 = 0$, where n_i is the number of the needles. We do not find a simple rule for the number of dots.

- Law generalization: $c_{L,4} = 0$, yielding $n_4 = 0$, and the coordination (?,0).
- Pulling back to the original space, we shall see an image with an arbitrary number of dots but no needles.
- Validity check: only one image corresponds to this description ✓.

Finally, case (d) reads:

- Base space description ('n' means needle, "d" is a dot and "e" is an empty dot): dnnn, dn–en and e–e–dn.
- We use the number of the constituents as before: (1,0,3), (1,1,2) and (1,2,1).
- Law: here, we find a lot of simple rules: $c_{L1,i} = d_i - 1 = 0$, $c_{L2,i} = n_{i+1} - n_i + 1 = 1$ and $c_{L2,i} = e_{i+1} - e_i - 1 = 0$.
- Law generalization: assuming that all laws are valid, we expect to see $d_4 = 1$, $e_4 = 3$ and $n_4 = 0$. This corresponds to the coordination (1,3,0).
- Pulling back to the original space, we shall see an image with one dot, three empty dots and zero needles.
- Validity check: only one image corresponds to this description ✓.

It is worth recalling two aspects of the task d). First, it involves the observation of a law "similar" to one used in a previously solved task. Either we would say the counting replaced by a count down or the problem solver can find the law that has parameter p, which can and should be adjusted according to the task. The second observation is more abstract. One can see that (1,3,0) is a solution of the task, while someone else may say that the task is "continue the sequence", and, in that case, (1,3,0) is next in the sequence but cannot be continued given that we have no "negative needles". There can be two different tasks or two different task contexts. Either the task is "choose the right figure" or "continue the series". Alternatively, the task context can be a selection or sequences, depending on the formalization of the task. Finally, these possible alternatives also show that two different tasks may have seemingly the same solution, while their abstraction level and task definitions are different.

There is a different aspect of these problems, namely, what happens if we establish simple rules, but there is no option that satisfies all rules, or there are several such options. These problems lead to a more subtle analysis that we postpone to future studies.

6. Extension

Imagine that we have Ω (or Ω_1), the grand universe (or a relevant subset), but it is not able to exhibit some phenomena that we would like to capture. Such expectation is coded very much in our brain, our sensual experiences, models, concepts, abstractions and generalizations that we previously acquired. In that case, we are tempted to *extend* Ω to $\Omega^+ \supset \Omega$, which can incorporate these expectations.

Let us emphasize that, by extension, we create something new, which recovers phenomena that in part were present in the original scenario but gain a better description in the new, wider one. In the better description, we refer again to the context, which evaluates the result of the extension, and will approve it if it is found more consistent, simple or even simpler than the original, giving it stronger explanatory power, etc.

Let us provide some examples in order to convince the reader that our approach to the extension has sense. Our first and foremost example is the evolution of our sense and understanding of numbers. Initially, we only had positive integers. Then, the invention of zero and negative numbers occurred. Both are necessary to maintain the framework for number subtraction and possess the same algebraic properties for addition and subtraction.

The Greeks were not acquainted with rational numbers, which are needed to keep the results of division within a framework while preserving the algebraic properties of addition, subtraction, multiplication, and division. Subsequently, we incorporated irra-

tional numbers, transcendental numbers, imaginary numbers and quaternions, all while maintaining the already established algebraic properties.

Irrational numbers became essential for finding the square root of integers and as roots of polynomials. Transcendental numbers like π and e initially appeared sporadically before finding their place and completing the set of real numbers. The introduction of the square root of negative numbers was necessary to have a set of numbers in which negatives have roots and to include roots of all kinds of polynomials. Quaternions represent a further formal extension of imaginary numbers while retaining all the properties of previous number systems, proving useful in theoretical physics.

A similar evolutionary extension can be observed in other branches of mathematics, such as geometry, as well as in other fields of knowledge.

Color theory explains how colors interact, harmonize and influence the audience's sensation. This extension of the numerous observed effects has two crucial characteristics. Firstly, the impact has changed throughout history, revealing how different human perceptions may arise when exposed to the same or similar visual effects. Secondly, this theory is unrelated to the spectral nature of light that produces colors. Ultimately, our perception of colors results from the superposition of signals from cones sensitive to different basic colors. For instance, the mixture of blue and yellow appears as green, although the wavelengths of blue and yellow do not change or mix. This example underscores the importance of the context in which the extension takes place.

Color theory operates with the names of colors and their associated emotions, compiling them into a more general framework. In contrast, in physics, wavelength, a specific property, provides a new unified framework for explaining colors.

After those examples, let us coin a formal framework of extension. Let us consider the universe Ω and functions on it (or on its power). Let us assume that we have a subset Ω_1 (or the direct power of it for multivariate functions).

We have a function $f : \Omega_1 \to R$ describing different tasks, depending on R (e.g., for clustering, R is a finite set, for transformations, $R = \Omega$ and, for a law, R is true or false). We may expect that some invariance is satisfied by f, i.e., there is a $b : \Omega_1 \to \Omega_1$ map, which is not a bijection; for that, $f = f \circ b$ in the appropriate domain. We may have no meaningful extension of f to the full Ω. Our expectation towards the extension is that on Ω^+, our extended function $f^+ : \Omega^+ \to R^+$ keeps the properties of the original f. Moreover, it shall be the same in the original domain. Formally, if π is the projection of Ω^+ onto Ω and ψ is that of R^+ onto R, then for all ω^+

$$\psi(f^+(\omega^+)) = f(\pi(\omega^+)) \tag{4}$$

Let Ω' be another set (universe), and let $\Omega^+ = \Omega \times \Omega'$. Similarly, $R^+ = R \times R'$. The crucial choice is, of course, Ω', R' and a new function $f' : \Omega' \to R'$. Consider the coordinate system C of Ω, and extend it with new coordinates $C' = \{c_{N+1}, \ldots, c_{N+M}\}$. The new coordinate system is C^+, the joint of C and C'.

We extend f in a straightforward way $f^+ : \Omega^+ \to R^+$,$R^+ = R \times \Omega'$ Let $\omega^+ = (\omega, \omega') \in \Omega^+$ and

$$f^+((\omega, \omega')) = (f(\omega), f'(\omega'))$$

The result is a completed commutative diagram depicted on Figure 3.

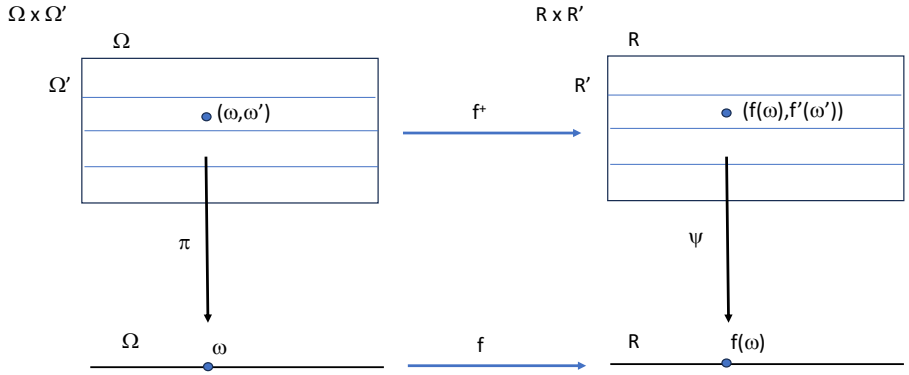

Figure 3. The commutative diagram of the extension.

We owe to the reader with the demonstration of how the above abstract scheme of extension works in practice.

From our previous examples, the most evident ones are the negative numbers and the imaginary numbers. Let $\Omega' = \{0, 1\}$ in which 0 represents the non-negative numbers and 1 the negatives (similarly, the real and imaginary numbers, respectively). f the subtraction acts on pairs on the extended set of numbers and keeps the nice properties (like associativity).

The example of the improvement of a classification scheme can be the animal taxonomy. Early classifications are based on size, shape, the number of legs, body cover (fur, feather, etc.) and living environment (water, air and ground). The schemes have some nice invariance properties. Most of the animals that live in the air can fly, have feathers and are birds, but there are a few exceptions, e.g., bats. If the feature set is extended to include genes, all the inconsistencies disappear, and even the evolutionary pathways can be discovered. A consistent new structure can be obtained.

In the classification context, we have Ω and a partition function $f : \Omega \to \{1, 2, \ldots M\}$. For a better classification, in this case, the extension needs new features, coordinates, but the extension of Ω is not needed (except that we found new items to incorporate that are not fitting into a simple set enlargement solution). Based on a better set of features, a new classification is possible, which has better explanatory power, fit to a goal/context given prior to the solution of the classification. As a special case, the new classification produces equivalent classes. The new classification may provide a better model and may produce not just one classification but a hierarchy of nested classes, a tree structure, like the evolutionary tree based on the sequencing of genes.

Let us go into the various outcomes that extension can offer. As mentioned earlier, when the object space is not expanded, extension still may involve refining the classification. We introduce new coordinates (features), leading to the splitting of some classes. It is also possible that certain features go unused, resulting in a complete redefinition of the classes. In a way, the latter scenario is not a straightforward extension; although it starts with an expansion of the feature set, it ultimately reduces it as well. However, it is clear that it begins with an extension step and contributes to a more accurate description (understanding) of Ω and proves to be beneficial.

In cases where new items (species) need to be added to Ω, it becomes clear that an extension of the coordinate set is necessary to include enough descriptive coordinates. Even if some of the descriptive attributes remain unused, the new items may exhibit properties that are worth describing, prompting an expansion of the coordinate set.

Let us now revisit situations in which it becomes apparent that our initial classification is incorrect, as exemplified by the case of inaccurately classifying bats as birds (see the first column in Figure 4 for the features and the last for the class assignment).

Ω	can fly	has feather	has wings	abstraction to birds
robin	1	1	1	1
eagle	1	1	1	1
ostrich	0	1	1	0/1
bat	1	0	1	1/0
rabbit	0	0	0	0

Figure 4. The extension in classification context. The original idea of classification (bird or not bird) is based on the 'can fly' feature of the animals (red box). This is not sufficient to classify the objects correctly, and we need an extension of the feature set, that leads to a better classification.

Let us assume that we have several animals in our sample, some of which can fly, such as bats, and others that cannot, like ostriches and rabbits, which lack wings. If we only use the feature "can fly," all animals except ostriches and rabbits are grouped into one class, and the abstraction of that class might mistakenly be considered as birds. This results in classifying bats as birds and ostriches as non-birds.

If we include further features like "has feathers" and "has wings", we achieve a much better classification. However, even the inclusion of "has wings" alone is not sufficient. In the left-most column of Figure 4, we indicated the incorrect/good classification.

It is clear that our definition does not provide a receipt for the extension, only a framework for it. That is a major challenge, the point where intuition, analogy and other mental schemes contribute to the extension. Those guides to extension are external with respect to the ground universe. There are many efforts to formalize them and use them in AI. We postpone our attempt in that direction to a next paper.

7. Conclusions

For the development of artificial intelligence, it is inevitable to understand the points where natural intelligence differs from the current AI solution. As we argue in this paper, the most robust difference is that mainstream AI solutions use a System-1-type approach, which is strongly connected to the (analytic) classification task. Human intelligence, however, extensively uses the System 2 approach, where structured information can be maintained and used.

The mathematical basis of the System 2 models lies in the coordination of the observed world. This covers the representation of objects, but also actions can be represented in this way. The sensors or sensational capacities of the intelligent actor provide a raw coordination. However, not all are important in a given context, i.e., regarding certain action sets. The first task of intelligence is to distill those coordinates that are relevant in that context. We gave the definition of relevance in Definition 1. In words, the relevant features of a quantity are those that are constant over the class singled out by the given quantity. In other words, the relevant features are the laws of the given quantity.

In System 1, we choose features as the desired quantities themselves. This is a viable approach, but it has some disadvantages, discussed in the text, for example, the specificity of the features to the quantities.

System 2 use more general relevant features that are reusable in different contexts. We discuss in the text how we shall establish these general features based on examples

(that may be corrupted by false examples) and the symmetries of the given quantities. Only with these general relevant features can we speak about the concepts like abstraction, generalization and extension, which are basic elements in human thinking.

Abstraction is nothing else than augmenting the observed world by the relevant features (i.e., coordinates) we found. Usually we are not even aware that raw data information (e.g., we do not have direct access to the retina information), rather we exclusively use the abstracted features. If we consider the relevant features of several quantities, then we shall prefer those that are common in them—this is the generalization of concepts over different quantities. Then, it is conceivable that certain coordinate combinations do not exist in the real world. Some of them may be so important (like the negative numbers) that we can consider to extend the world to incorporate these combinations.

None of these concepts are available in the System 1 approach; on the other hand, they provide the very basis of human thinking. The main message of this paper is that in order to build real thinking machines, we shall provide relevant features based on the observed laws of the real world. Only with these abstracted and generalized concepts is it possible to model the relevant part of our environment.

Of course, in the present paper, we could only sketch the mathematical background, so a lot of questions remain to be answered.

Author Contributions: Conceptualization, A.J. and A.T.; Formal analysis, A.J. and A.T.; Resources, A.T.; Writing—original draft, A.J. and A.T.; Writing—review & editing, A.T.; Project administration, A.J. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Hungarian Government and the European Union in the framework of a Grant Agreement No. MILAB RRF-2.3.1-21-2022-00004.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kahneman, D. *Thinking, Fast and Slow*, 1st ed.; Farrar, Straus and Giroux: New York, NY, USA, 2013.
2. Carroll, J.B. The three-stratum theory of cognitive abilities. In *Contemporary Intellectual Assessment: Theories, Tests, and Issues*; Flanagan, D.P., Genshaft, J.L., Harrison, P.L., Eds.; Guilford Press: New York, NY, USA, 1997.
3. Turing, A. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–460. . [CrossRef]
4. Neufeld, E.; Finnestad, S. In defense of the Turing test. *AI & Soc.* **2020**, *35*, 819–827. [CrossRef]
5. Jakovac, A.; Berenyi, D.; Posfay, P. Understanding understanding: A renormalization group inspired model of (artificial) intelligence. *arXiv* **2020**, arXiv:2010.13482.
6. Chollet, F. The limitations of deep learning. In *Deep Learning with Python*; Manning Publications Co.: Shelter Island, NY, USA, 2017.
7. Jiang, Y.; Rajendran, G.; Ravikumar, P.; Aragam, B.; Veitch, V. On the Origins of Linear Representations in Large Language Models. *arXiv* **2024**, arXiv:2403.03867
8. Park, C.F.; Lee, A.; Lubana, E.S.; Yang, Y.; Okawa, M.; Nishi, K.; Wattenberg, M.; Tanaka, H. ICLR: In-Context Learning of Representations. *arXiv* **2025**, arXiv:2501.00070.
9. Mitchell, M. Abstraction and Analogy-Making in Artificial Intelligence. *arXiv* **2021**, arXiv:2102.10717. [CrossRef] [PubMed]
10. Gong, F.; Xia, Z. Interpreting Deep Learning by Establishing a Rigorous Corresponding Relationship with Renormalization Group. *arXiv* **2022**, arXiv:2212.00005.
11. Holte, R.C.; Choueiry, B.Y. Abstraction and reformulation in artificial intelligence. *Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci.* **2003**, *358*, 1197–1204. [CrossRef] [PubMed]
12. Ponsen, M.; Taylor, M.E.; Tuyls, K. Abstraction and generalization in reinforcement learning: A summary and framework. In *International Workshop on Adaptive and Learning Agents*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–32.
13. Aubier, T.G.; Sherratt, T.N. State-dependent decision-making by predators and its consequences for mimicry. *Am. Nat.* **2020**, *196*, E127–E144. [CrossRef] [PubMed]
14. Biró, T.S.; Jakovác, A. Entropy of Artificial Intelligence. *Universe* **2022**, *8*, 53. [CrossRef]
15. Jakovác, A.; Telcs, A. A Note on Representational Understanding. *Entropy* **2022**, *24*, 1313. [CrossRef] [PubMed]

16. Yuan, B.; Zhang, J.; Lyu, A.; Wu, J.; Wang, Z.; Yang, M.; Liu, K.; Mou, M.; Cui, P. Emergence and causality in complex systems: A survey of causal emergence and related quantitative studies. *Entropy* **2024**, *26*, 108. [CrossRef] [PubMed]
17. Bäckström, C.; Jonsson, P. A framework for analysing state-abstraction methods. *Artif. Intell.* **2022**, *302*, 103608. [CrossRef]
18. Vowels, M.J.; Camgoz, N.C.; Bowden, R. D'ya like dags? A survey on structure learning and causal discovery. *ACM Comput. Surv.* **2022**, *55*, 82. [CrossRef]
19. Hale, B.; Wright, C. The metaontology of abstraction. In *Metametaphysics: New Essays on the Foundations of Ontology*; Oxford University Press: Oxford, UK, 2009; Volume 529, pp. 178–212.
20. Zhu, S.P.; Wang, L.; Luo, C.; Correia, J.A.; De Jesus, A.M.; Berto, F.; Wang, Q.Y. Physics-informed machine learning and its structural integrity applications: State of the art. *Philos. Trans. R. Soc. A* **2023**, *381*, 20220406. [CrossRef] [PubMed]