

JYX



**This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.**

**Author(s):** Saariluoma, Pertti; Karvonen, Antero

**Title:** Theory languages in designing artificial intelligence

**Year:** 2023

**Version:** Published version

**Copyright:** © The Author(s) 2023

**Rights:** CC BY 4.0


**Rights url:** <https://creativecommons.org/licenses/by/4.0/>

**Please cite the original version:**

Saariluoma, P., & Karvonen, A. (2023). Theory languages in designing artificial intelligence. *AI and Society*, Early online. <https://doi.org/10.1007/s00146-023-01716-y>



# Theory languages in designing artificial intelligence

Pertti Saariluoma<sup>1</sup> · Antero Karvonen<sup>2</sup> 

Received: 11 October 2022 / Accepted: 21 June 2023

© The Author(s) 2023

## Abstract

The foundations of AI design discourse are worth analyzing. Here, attention is paid to the nature of theory languages used in designing new AI technologies because the limits of these languages can clarify some fundamental questions in the development of AI. We discuss three types of theory language used in designing AI products: formal, computational, and natural. Formal languages, such as mathematics, logic, and programming languages, have fixed meanings and no actual-world semantics. They are context- and practically content-free. Computational languages use terms referring to the actual world, i.e., to entities, events, and thoughts. Thus, computational languages have actual-world references and semantics. They are thus no longer context- or content-free. However, computational languages always have fixed meanings and, for this reason, limited domains of reference. Finally, unlike formal and computational languages, natural languages are creative, dynamic, and productive. Consequently, they can refer to an unlimited number of objects and their attributes in an unlimited number of domains. The differences between the three theory languages enable us to reflect on the traditional problems of strong and weak AI.

**Keywords** Artificial intelligence · Theory languages · Formal languages · Computational languages · Natural languages

## 1 Introduction

Traditional electromechanical technologies are built on innovative ideas concerning either energy or materials (Bernal 1969; Derry and Williams 1960; Wiener 1948). Future intelligent information technologies will be characterized by intelligent information processing (Ford 2014; Fukuda 2020; Tegmark 2017) and will accomplish numerous tasks previously requiring human involvement (Bringsjord and Govindarajulu 2020; Minsky 1967; Newell and Simon 1972; Russell and Norvig 1995). Consequently, these new technologies will replace people in various areas. For instance, in the near future, autonomous cars, ships, and aircraft will transport goods with minimal proximal human effort. Ultimately, intelligent and autonomous systems may become

ubiquitous, following people from the cradle to the grave (Ford 2014; Fukuda 2020; Gungel 2012; Tegmark 2017).

An important instrument in the ideation and application of intelligent technologies is theory languages, particularly their capacity to determine the scope and limits of intelligent technologies in contemporary society (Saariluoma 1997). All new ideas and thoughts must be expressed and explicated, and for this reason, the theory languages used in design work are crucial as they impose limits on researchers' and designers' thinking (Kant 1781/1976; Kuhn 1962; Saariluoma 1997; Wittgenstein 1921/1961).

The foundations of all learning and intelligent thinking are engraved in the concepts used by researchers and designers. The conceptual limitations of a theory language are demarcated by observations, facts, theories, and design ideals (Saariluoma 1997; Saariluoma et al. 2016). The foundations of science-based activities entail both an explicit and tacit understanding of life and the world. The limits of theory languages restrict hypotheses and truths as well as the intuitive presuppositions underlying researchers' thinking (Saariluoma 1997). A way forward in research and design is to explicate these presuppositions and thereby further clarify how researchers think (Wittgenstein 1953). This kind of

---

✉ Pertti Saariluoma  
ps@jyu.fi

Antero Karvonen  
antero.karvonen@vtt.fi

<sup>1</sup> (Faculty of Information Technology, Cognitive Science),  
University of Jyväskylä, Jyväskylä, Finland

<sup>2</sup> VTT Technical Research Center of Finland, Espoo, Finland

reflective scientific activity is called foundational analysis (Saariluoma 1997).

The main characteristic of an intelligent society is captured in the concept of intelligent information processing (Floridi 2011, 2013; Newell and Simon 1972; Saariluoma et al. 2016). People encode and process information, but many technical artefacts can also process information (Newell and Simon 1972; Turing 1936–1937, 1950). Therefore, it is logical to study the theory languages used in representing information in the human mind, in machines, and in their collaborative processes.

Digital intelligence and intelligent technologies are grounded on a single but versatile notion: information (Floridi 2011, 2013; Newell and Simon 1972; Wiener 1948). Both technological and human intelligence is possible insofar as it is possible to represent reality and to manipulate these representations (Petzold 2008; Turing 1936–1937, 1950). Thus, it is also possible to imitate states of affairs and even to hypothetically analyze possible states. Such representation manipulation processes have been called information processing (Floridi 2011, 2013; Kåhre 2002; Lindsay and Norman 1977; Newell and Simon 1972).

According to Wiener (1948), information is neither matter nor energy but instead belongs to its own ontological domain. It can be said that our era is one that, to a significant degree, emerged from the scientific conceptualization of information around the time of the Second World War (Aspray 1985; Waldrop 2018). This conceptualization served as the basis for mathematical communication theory, computer science, and AI (Russell and Norvig 1995; Shannon 1948). Through the notion of the mind as an information processing system, information processing also constitutes the core of cognitive science (Floridi 2011; Newell and Simon 1972; Turing 1950). While the electromechanical substrate of AI is a necessary and important domain of human innovation, it is also clear that the construction and development of information processes that accomplish tasks are equally, if not more, important.

## 1.1 Aspects of information

Information stands for something and represents various states of affairs. Information can be approached from different conceptual perspectives. The most common conceptual system used in investigating information is the mathematical theory of information or communication. This theory was developed by Shannon (1948) in the late 1940s (Floridi 2011, 2013). The core problem of traditional information theory was determining how to measure the *amount* of information (Kåhre 2002; Shannon 1948; Shannon and Weaver 1949).

The original concept of information as a quantitative measure is a powerful notion in technical design but is

unsatisfactory with respect to other aspects of information. Quantity cannot express what information is about, or its quality, and for this reason many researchers have sought to extend the original idea by studying the semantic aspects of information (Bar-Hillel and Carnap 1953; Floridi 2013; Hintikka 1973). Semantic information references reality and as such can express actual-world states instead of mere symbols.

As semantic information can express actual-world states, it has, consequently, truth values (Bar-Hillel and Carnap 1953; Floridi 2013). When the focus is only on the amount of information, it is not easy to determine whether the information is true or false, because one must know the content of the information before it becomes possible to decode its truth value. The two main conceptions of information in cognitive science have been thus far metrical and semantic. Here, we will extend the analysis of information to the information content of messages or to mental content (Myllylä and Saariluoma 2022; Saariluoma 1997, 2001; Saariluoma and Rousi 2015). Thus, instead of asking how many bits it takes to convey “My Aunt is a lunatic”, we ask what this information means in the minds of the people to whom it is communicated. We especially, we focus on the relevant information contents in mental representations.

## 1.2 Turing’s legacy

Ever since Aristotle (1984), at least, it has been a common practice to abstract the formal structures of representations. Instead of analyzing representations on the concrete level, e.g., “All human beings are mortal”, “Socrates is a human being”, and thus “Socrates is mortal”, one can study the general structure “All A are M, S is A, and thus S is M”. Abstraction makes the representation general, and by substituting concrete information with the variables A, M, and S, one can build an unlimited number of respective inference patterns. Abstraction extracts a pattern from a concrete context by removing its content. The very notion of abstraction implies both what is lost and what is gained. What is gained is universality and what is lost is particularity. This seems almost painfully obvious, but the implication, in terms of AI and its fundamental limitations, is significant.

Aristotle’s idea was developed further in the works of Leibniz (von Wright 1956). He understood that a step forward could be the formalization of inference operations or transformation rules. He presented the idea that one could generate one representation from another by deduction (Passmore 1957; von Wright 1956). This means that abstract representations could be transformed into new

representations by following inference rules, like moves on a chessboard. Leibniz thus created the idea of calculus,<sup>1</sup> a formal system whereby scientific and even metaphysical issues could be settled based on the mechanical manipulation of symbols alone, or the *sc. characteristic universalis* (von Wright 1956). This fundamental idea later became the foundation of formal information processing thanks to Alan Turing (1935, 1936, 1950).

Turing's thinking demonstrated that it was possible to manipulate information on artificial substrates that possessed a quasi-autonomous character or independence from immediate human involvement. One representation can be transformed into another—the insight, which made deciphering the Enigma possible (Hodges 2014). There is not necessarily anything qualitatively new in conclusions compared to information in premises. These kinds of inferential processes take place in the thinking human mind (Hodges 2014; Newell and Simon 1972; Turing 1950). Transformation processes can be called, in some sense, intelligent, and digital intelligence in particular is based on the possibility of representing information and manipulating or processing such representations into other representations. This is a significant step forward, as these tasks were previously performed only by the human mind. While it is clear that this type of theorizing does not as such address the *origin* of premises, representations, or the creative character of thought more broadly, this omission does not undermine its achievements.

The key feature of digital intelligence can be found in its fundamental discreteness and clarity, which manifests in, for example, stepwise time, discrete symbols, and syntactical rules (Anderson 1993; Newell and Simon 1972). It is based on the possibility of representing information and manipulating or processing such representations into other representations, in such a way that a change in the mechanical processor becomes coincident with a change in the representation. The human mind, as it were, extends itself outward to the world and maps an aspect of itself (an information process) on physical causal processes. In this process, the human mind organizes the physical system into a mechanism that corresponds to the information process. The coherence between representations and physical processes is constructed by the human mind using relevant conceptual languages (Saariluoma 1997). The level of human involvement needed (and where it is needed) depends partially on the developmental stage of technology. Today, aspects of computer programming, such as assembly, have become so well-specified that they can be automated.

In practice, information-processing machines can accomplish many tasks typical of human beings. Even within their

foundational limitations, originating as numerical calculators, intelligent information processing systems have become extremely powerful (Boden 2016; Russell and Norvig 1995). This is why it makes sense to consider the foundations of representation languages when thinking about intelligent information technologies.

## 2 Theory languages for processing information

Floridi (2011, 2013) pointed out how abstraction levels matter in our thinking about information processing. The main question in this regard is what the theory languages of abstraction are and how these languages affect the innovative thinking used in creating an intelligent society.

Theory languages are discourses or language games that researchers use when they investigate some specific problem (or theme) and design technologies to solve this problem (Foucault 1972; Habermas 1981; Wittgenstein 1953). Thus, set theory, calculus, and topologies are common theory languages used to investigate the relevant problems of mathematics. Neuroergonomics, Kansei engineering, and usability are examples of human-technology interaction (HTI) design languages (Saariluoma et al. 2016).

Theory languages are limited in their capacity to express problem-relevant information (Saariluoma 1997). Thus, historically, behaviorism in psychology excluded attention, memory, and even thinking from the focal areas of psychological research and, for this reason, it could not investigate, for example, human information processing (Lindsay and Norman 1977; Watson 1914).

In creating intelligent technological systems, researchers and designers need domain- and problem-specific theory languages. Here, we will focus on three different theory language types important in constructing intelligent technologies: formal languages (Kleene 1971; Salomaa 1985; Turing 1936–1937), computational languages (Avison and Fitzgerald 1995; Boden 2016; Harnad 1990), and natural languages (Carstairs-McCarthy 2001; Chrystal 1971; Lyons 1977, 1995; de Saussure 1983).

### 2.1 Formal languages—their scope and limits

By *formal languages*, we mean symbolic languages that do not have specified and concrete references to the world and its entities, events, or ideas. The symbols, syntactic rules, and expressions of formal languages are content- and context-free symbolic constructions. On the other hand, one cannot claim that formal expressions are necessarily arbitrary or randomly irrational. Rather, they are symbolic constructions with their own construction rules (Kleene 1971; Salomaa 1985). “The world” of abstract formalisms has a

<sup>1</sup> In the original sense, not the branch of mathematics Leibniz developed along with Newton.

kind of pristine clarity and strict lawfulness that it seemingly achieves by “rising above” particular facts of the matter. Typical examples of formal languages are mathematics, formal logic, and general-purpose programming languages such as C#, Lisp, or Python.

It may be too extreme to say that formal logic or programming languages, for example, have no interpretation or meanings. One may claim that these languages have *minimal* interpretation, and that one goal of the formal approach is to discover valid, algorithmic, and syntactic processes for reasoning that do not refer to the content of the expressions beyond the necessary minimum. Strictly speaking, programming languages have their references in machine states and memories. However, they do not have any other real-world references, unless such references are constructed (turning them into what we call computational languages).

Formal theory languages have syntactic properties (Salomaa 1985). The process of creating a formal language is thus based on creating a system of formal symbols and a formal axiomatic. Syntax defines the conditions for well-formed representations and possible rules for their transformation (Kleene 1971; Salomaa 1985). Operations in a formal theory language, such as *concatenation*, are largely content-free (Salomaa 1985).

In practice, formal languages can have semantics referring to sets with members, or so-called model sets. Then, the truth of a formal expression can be defined on the grounds of a reference to the model set. Expressions are true if the state of affairs they define is true in the defined model set. However, members of the model set do not have actual-world references. They form a section of the actual world, but this section is too abstract to be directly used in actual-world actions.

One can express the intuitive ideas above more systematically by stating that a formal language has.

1. a set of alphabets as words, which do not yet have references to the actual world
2. a set of rules defining how symbols can be syntactically combined
3. a set of rewriting rules outlining how one can derive from a set of symbols another set of symbols

Additionally, the following rules apply to expressions in a formal language:

4. The semantics of formal expressions can be defined in a model set that has elements, but the elements do not have concrete actual-world references.
5. The truth of a formal expression is defined on the grounds of the prevailing states of affairs in the model sets.
6. The expressions of formal languages need only have abstract semantics of truth values.

Formal languages are explicitly abstract and essentially context- and content-free. Thus, one can study the abstract properties of expressions without binding these expressions to the specific and concrete properties of any actual-world contexts—physical, behavioral, or social. Elementary sentences refer to any object or any state of affairs, and this has the consequence that one cannot use formal languages to separate concrete objects. On the formal level, people, potatoes, or legal statements are equal, although in the actual world, they are very different. Abstraction makes expressions minimal in content, and for this reason, formal languages have problems expressing important actual-world concepts. The expressions of formal languages can be senseful or senseless, and as such one cannot ground all of the intelligence on any formal language.

Formal languages cannot be used as the sole grounding of intelligent information systems. Moreover, formal language expressions cannot be used to solve problems of *relevance*. Relevance refers to meanings and to which elements belong to one definable group or another. Relevance is a mechanism that defines a set of meaningful and meaningless items. This operation is not possible in any formal language. The key point is that it is impossible strictly from *within* the language (Gödel 1962; Rosen 1991). One can combinatorically divide any model set in a number of ways, but there is no formal-level possibility to decide what or which of the divisions is better than the other combinatorial possibilities. Therefore, it is essential to shift our attention to partially interpreted languages.

## 2.2 Computational languages

Formal languages such as formal logic or general-purpose programming languages are content-free, and they cannot be connected with the actual world, but their expressions and truth values have been defined by means of model sets (Hintikka 1973). However, it is possible to construct meaningful actual-world semantics for formal languages. Their syntactic structures and terms can be associated with actual-world entities, relations, and events (Minsky 1968).

Theory languages that have actual-world references and semantics are called here *computational languages*. We discuss computational here in a wider sense, of course than mere numerical calculation. Examples of computational languages are languages of any system of signs in which terms refer to the actual world and thus have actual-world references. Simulation models and many engineering design languages, such as digital twins, and human digital twins provide relevant examples (Boden 2016; Jones et al. 2020; Saariluoma et al. 2021). Calculating provides an interesting example. Consider the following sentence: “Two apples and three oranges make five fruits.” Is this a formal, computational, or natural language statement? On the face of it,

it must be *in* natural language. But it *contains* a computational language that combines a real-world reference and a mathematical process that gives the statement coherence. Formal it is not, although it could be *expressed* formally by evacuating the content and references. A further point is that computational languages are referentially individuated based on a functional purpose: digital twins, for example, have a purpose for their references (which are selected for the computation) and the calculating example could have a functional purpose of making the process more intuitive for a child or relevant for the purposes of a shopkeeper.

Computational languages are different from formal languages, such as programming languages, that are used to craft them. They are on a linked but separate level. A programming language refers to machine operations, strictly speaking. A computer program is typically built on the basis of a programming language. However, by way of human use, a computer program smuggles in a reference to the actual world either tacitly or explicitly. This aspect of programs and programming—a part of what we refer to as computational language—seems to have rarely been noticed explicitly.

Computational languages are built on formal languages by giving interpretations i.e. defining references in the actual world to formal symbols and their relations. The references can be imagined objects and things, as in games. They can be physical states encoded by sensory systems. They always have unique references in the actual world. However, the references are fixed and they cannot vary on the spot. Of course, it is possible in neural networks to modify inputs and intermediate symbols, but they have static meanings. The fundamental point is that while formal languages are content-free, computational languages have content *given* to them, but they do not have the power to generate or create content (Searle 1991).

A crucial difference between formal and computational languages is found in interpretations, meaning-giving, or symbol grounding (Harnad 1990). By meaning giving or symbol grounding, we refer to the process by which symbols and syntactic structures are associated with actual-world entities, events, or ideas. In formal languages, symbols or syntactic operations do not have any references to aspects of the actual world but to model sets only (Hintikka 1963, 1973; Kleene 1971; Salomaa 1985). In principle, they *can* have such references, as the logical symbol “A” or the number “*n*” can refer to all of the entities of the actual world, and the number “*n*” can also refer to all of the entities whose number is *n*, but they do not have any concrete content.

Arithmetic operations can be organized into functions that, for instance, make a system move a knight in chess, calculate a statistical analysis, or imitate how turbines work. If computational operations are not able to carry out the required process, they are not valid in the actual world to which they refer. The information content gives

computational languages new properties. Computational theory languages present the possibility of asking questions about meaning, truth, accuracy, and relevance. One can study whether the terms of a computational language refer to how things are in the actual world, and one can ask if the behavior of a computational model is correct; the actual world to which it refers corresponds to some level of the model, be it output or steps toward it. It is again a different but related question whether the model has an analogue in the actual world—whether it approximates the actual world or merely accurately predicts it through a different kind of structure. Note, however, that these questions cannot be answered from “within” the model but must be evaluated by the powers of the human mind vis-à-vis the actual world. This highlights the question of whether the model (and the language) is relevant for the problem to be solved. A physicist is not primarily interested in whether the model predicts, but instead, whether it predicts *and* is an analogue to the phenomenon under study. In AI, this is not necessarily the case: Results and performance are the core issues today—although historically, AI models were simultaneously models of the mind (Newell and Simon 1972).

To summarize:

1. Computational theory language terms are interpreted, and thus they have references in defined aspects of the actual world.
2. Computational language operations have respective actions in the actual world.
3. The truth of expressions and representations can be defined by comparing the computational language expressions with corresponding states in the actual world.
4. It makes sense to ask whether terms and operations are correctly defined.
5. One can ask whether or not the given system is relevant, although its relevance cannot be defined in computational language.
6. Computational languages have fixed semantics.

Computational languages have specific features, which limit their use. The most important of these features can be called static or fixed meanings. A term in a computational language has some specified and unique reference. This property is an important difference from natural languages. The meanings of terms in computational languages are defined in a unique manner. They can have some sensory value or some variable system of values, but they will nonetheless be fixed.

Another property of computational languages is their limited number of terms. These languages cannot generate an infinite number of new concepts and terms on the spot on the

grounds of new experiences. One could say that computational languages are not productive, which is a key property of natural languages (Lyons 1977). Therefore, as computational languages cannot generate an infinite number of new words and other signs, they are static and limited compared to natural languages.

The problem of fixedness and the stable character of computational languages is not new. It was likely first noticed by Ludwig Wittgenstein (1953) in his sc. late philosophy. In this philosophy, he demonstrated the problematic character of defining terms, which have, at best, a family resemblance, in that one interpretation of the word *game*, e.g., children playing a board game, has very little in common with playing NFL football or with partisan political games. The meaning of a word is defined by how it is used (Wittgenstein 1953). Equally, the meaning of a concept changes according to how it is deployed: Politics can be thought of as a game, but also as theater. War can be thought of as a game, but also as an extension of politics. To improve the clarity of our analysis, we will consider natural languages more closely.

### 2.3 Natural languages

Natural languages refer to the ways in which people speak and communicate with each other. They are sign systems with a number of specific characteristics. There are at least 6000 distinctive natural languages, but they have common features, which make them similar as a means of communication, expression, and thinking (Leech 1983; Lyons 1977, 1995). To understand the function of natural languages in designing intelligent technologies, it is essential to consider some specific properties of natural languages compared to formal and computational languages.

Natural languages are in principle arbitrary; their words and signs can be morpho-syntactically and grammatically formed, and signs are discrete so that they can be differentiated from each other (Leech 1983; Lyons 1977). Even ambiguous terms with multiple references can be differentiated on the conceptual level by contextual information. Terms also have historical origins, and they have their arbitrariness, but it does not mean that any term can be used for any reference in practice. Natural languages are productive, as mentioned. This means that the speakers can generate and understand an unlimited number of expressions, and they can create new expressions on the spot. Finally, the expressions of natural language refer to their references through human mental representations and are as such expressions of human thoughts (Lyons 1977, 1995).

Natural languages have subsystems. One can call them, for example, social dialects, language games, or discourses (Habermas 1981, 2009; Wittgenstein 1953). These subsystems of natural languages often pertain to specific domains

and specific—although not necessarily fixed—sets of terms and meanings, and they are deployed by people who participate in and have experience with these domains and accept their associated discourse rules. Professionals such as aircraft pilots, paper engineers, psychologists, and economists live in their professional discourses.

The subsystems of natural languages are not strictly differentiated because they keep changing. Natural languages form linguistic platforms on which specific discourses are built. Thus, medical doctors can dictate sentences in which normal language expressions are made more precise by Latin names. In ICT (information and communication technology), one can use such metaphoric expressions as “agent” technology. Words as signs can be ambiguous and have multiple meanings. The conceptual contents evolve in use, and vague and metaphorical beginnings may result in precise and differentiated conceptual systems. In this sense, discourses (and the products thereof) evolve from premises and presuppositions that are fixed in a pragmatic sense, not in principle.

Natural languages are open and dynamically changing systems, and we can build an unlimited number of words and expressions in and through them. Every word is capable of having varying references. Thus, meanings in natural language are not fixed in the same way as they are in computational and formal languages. This openness makes natural languages creative. In principle though not always in practice, it is possible to find an expression for anything in human thoughts and in the actual world—including formal languages, as we are indeed doing here. Words as signs are symbols, which get meaning only when they are associated with concepts in the human mind. The meaning of meanings is in the contents of concepts (Ogden and Richards 1923). Thus, formal and computational languages are subsets of natural languages where, in the former, a domain has been abstracted from particularities, and, in the latter, abstractions are organized with respect to such particularities. “Traffic” between formal, computational, and natural languages is possible as they all originate in and are derived from the same natural language components, most broadly understood. Incidentally, these arguments can be seen to favor meaning and content as ontologically more basic than content-free formal operations.

The meanings of formal and computational languages are fixed, either in model sets or in some subdomain of the actual world. Thanks to the openness and creativity of natural languages, the problem of meanings in natural languages is somewhat different. The references of expressions can vary. Historically, atoms in, for example, Democritus’s time were different from atoms as described by Dalton, which in turn diverged from modern conceptions of atoms (Bernal 1969). The meanings of terms in natural languages and their sub-languages keep changing in the course of conceptual change and advancement (Saariluoma 1997).

One can ask what makes it possible for people to use natural languages in design thinking. Meanings in natural languages can be discerned through mental representations. Ogden and Richards (1923) argued that words refer to the actual world through concepts. Thus, concepts give meanings to words and expressions. In the same way, Chomsky (1957) and Fodor (1990) based their ideas concerning meaning on the notion of mental representations, and thus they left behind traditional behavior and other semantics, which saw meaning as a direct relation between signs and their actual-world references. Indeed, the dynamic and context-sensitive character of natural language makes sense once it is noted that it has an intimate and, once developed, almost automatic relation to other mental processes.

## 2.4 Meaning and mental representation

Information exists in signs and their organized collections or representations. These signs have references that can in a narrow sense also be called meanings. The use of an expression defines its meaning (Wittgenstein 1953). However, the references of signs, such as words or pictures, obtain their meaning content in human minds (Lyons 1977; Ogden and Richards 1923). The Rosetta Stone, for example, is a collection of hieroglyphic signs, but these signs had no known meanings at the time the stone was discovered and as such was not understood. They had references in principle, but their communicative value could not be much more than mere scrapes on the stone's surface until Champollion was able to interpret them using the non-hieroglyphic texts found on the same stone. Some references in forgotten languages may be impossible to ever know, such as customs, if no information is available. Other references are more universal, such as basic emotional states, which are widely accessible to most human beings. The broader point is that there are no references and no languages that we cannot, in principle, know due to their grounding in the human mind. There are of course countless references and meanings that an individual may never know in their lifetime, but this does not speak to the limitations of the language as such.

The necessity of mental content in defining the meanings of terms can be demonstrated by the *sc. world end thought experiment*. Thought experiments in general are instruments for concretizing conceptual problems in research (Kuehne 2005). In the "world end thought experiment," we assume that all of humankind has been destroyed. However, ants have been able to survive. Eventually, one ant enters the Metropolitan Museum of Art, ultimately walking on the surface of *Madame Ginoux*, painted by van Gogh. Would this ant be capable of understanding that it is walking not on a red book or a chair but instead on its painted representations? Only a human observer could make such a discrimination because ants do not write books or craft chairs due to

the limitations of their nervous and cultural systems. Thus, in the absence of human minds, information about books or chairs would be absent. This means that human information exists only insofar as a human being is present to decode it, whereas information comprehensible to an ant is of an entirely different kind.

Our ant would be able to communicate information about states of affairs with other ants by chemicals or movements. It would not, however, be capable of communicating human mental content because it would lack the faculties needed to mentally represent objects such as books, chairs, or paintings. It is therefore entirely justifiable to assert that human mental content is essential for defining human meanings. The consequence of previous arguments is that we have to separate the content of thoughts from the semantics of the language. Human thinking gives meanings to linguistic expressions.

## 2.5 Power of expression

The foundations of any kind of thinking are versatile. For example, the ways in which experiments are constructed necessarily limit what they can tell us about the human mind (Saariluoma 1997). Long ago, a mathematician was able to prove that there is no (natural) number that could express the ratio of the diagonal to the side of a square (Saariluoma 1997). This well-known Pythagorean case also reveals something more important than the idea that the world is not rational. There are mathematical problems that go beyond the power of expression of a particular theory language, i.e., the arithmetic of natural numbers. The solution to the problems posed by the Pythagorean sage was to develop a new theory language called *real numbers*. The limitations of theory languages do not concern mathematics alone as, for example, behaviorist social research cannot shed light on human information processing.

The power of expression is important here with respect to the scope and limits of the three presented theory languages. If some issue or state of affairs is outside the scope of a theory language, it cannot be faithfully represented in or through it. Thus, the actual world is outside and cannot be represented in formal languages. It is thus necessary to expand formal languages by associating them with the actual world and transforming them into computational languages.

Nevertheless, computational languages also have their limits. They are constructed on the grounds of formal languages, and they accordingly share some limits typical of formal languages. For instance, they have a limited set of terms and fixed meanings and are thus incapable of rendering an accurate representation of the actual world in general. They can operate only on the computational abstraction level.



The power of expression also lends a new understanding of the relations between the three theory languages. Formal languages are abstractions of natural and computational languages. Thus, they cannot express the concrete details of the actual world without interpretation. Interpretation makes it possible to create computational languages, which can refer to any aspect of the actual world. However, their fixed meanings do not allow them the creativity possessed by natural languages. Moving from natural languages to computational and formal languages in constructing representations makes the representations more abstract, but moving from formal to natural languages makes them more concrete and powerful.

### 3 Machine and human thinking

Because intelligent machines process information, the limits of theory languages have an interesting consequence. Specifically, the limits of the three theory languages presented here concerning their power of expression make it possible to scrutinize, from a new point of view, the classic problem of comparing human and machine languages. Machines can process information, but this means that their limits depend on those of the representational language on which they rely. Machines can only be as intelligent as the limits of their representational language allow.

All intelligent machines use computational languages in representing information. Therefore, the limits of computational languages determine the limits of machine intelligence. If a computational language does not have the capacity to express or construct representations, then the machines operating according to the language lack this capacity as well. Thus, the differences between natural languages and computational languages provide one interesting basis on which to discuss human and machine intelligence and their relative capacity to think.

One crucial limitation of computational languages is their fixed semantics. The world can vary indefinitely, and therefore natural languages have dynamic semantics. The references in natural languages are associated on the spot with symbols and ideas that convey meanings. The meanings of computational languages are defined by references when the alphabets and operations of these languages are constructed.

Computational representations can in principle describe any aspect of the real world. However, they are always limited in scope. The world, i.e., objects, events, actions, and thoughts, is not limited but can be infinitely complex. Even an object as simple as a coffee cup has endless attributes. Consequently, no AI system can generate unlimited representations of the world, just as no description of an infinite world can be included inside itself. Computational languages and their respective representations thus always constitute limited descriptions of the world, and accordingly,

no universal and general machine intelligence is possible. Any time the meaning of a term or an operation is fixed, some part of the world is excluded from the system.

However, one can consider the problem of human and machine intelligence from another perspective. AI systems, such as chess-playing programs, can surpass human performance, but they are useless in other domains. Similarly, it is possible to create computational languages and their respective computational representations for other domains. Thanks to the fixed semantics of these languages, they can be generalized only to a limited degree or extent and the type of fluid context-switching typical of human thought is difficult if not impossible to achieve. Furthermore, fixed semantics also mean that AI finds fixed *domains* or *environments* most tractable—for obvious reasons.

Human natural languages operate differently as they possess an unlimited number of symbols and can generate and define new symbols for any situation on the spot. Whereas computational languages obtain their semantics by definition and convention, natural languages acquire their semantics through concepts and human thoughts and, for this reason, these semantics are intrinsically dynamic.

As a matter of fact, computational terms acquire their meanings through natural languages, and their representations constitute subsets of natural language representations. Thanks to the dynamic nature of human thinking, natural language expressions obtain their content from the human mind and thus have no predefined limits. Of course, it is true that computational (and formal) languages also obtain their content (and other features) from human thought. The relationship here, however, is very different, as a natural language is part of the same immediate system as thought and is therefore by necessity and evolution highly related and complementary. In the external and artificial systems for which computational languages are used, this proximity is lost and various constraints are introduced. Thus, the fixed semantics of computational languages and associated representations make it impossible for AI to achieve either the generality or the versatility of human thinking. Furthermore, there at present no other basis for AI, given their grounding in formal and computational languages.

Of course, it is possible to build domains and problem-specific systems, which, in solving problems, can surpass human thinking. As there is no limit to the number of specific AI systems that can be constructed, it makes no sense to argue that AI, as a whole, could not surpass human performance, in all known specific domains. In such cases, it can be said that AI is “good enough” but nevertheless not sufficiently powerful to surpass human performance in all its generalities and manifestations. When operating in domains (or across domains) that are not specific, well-defined, or simple, human intelligence will remain crucial.

**Author contributions** Both authors contributed to the writing, proof-reading, and ideation of the article.

**Funding** Open Access funding provided by University of Jyväskylä (JYU). This research has been supported by Business Finland for the SEED-project (<https://seedecosystem.fi>).

**Availability of data and materials** Not applicable.

**Code availability** Not applicable.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

**Additional declarations for articles in life science journals that report the results of studies involving humans and/or animals** Not applicable.

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Anderson JR (1993) Rules of the mind. Erlbaum, Hillsdale
- Aristotle (1984) Complete works of Aristotle. Princeton University Press, Princeton
- Aspray WF (1985) The scientific conceptualization of information: a survey. *Ann History Comput* 7(2):117–140
- Avison D, Fitzgerald G (1995) Information systems development: methodologies, techniques and tools. McGraw-Hill, London
- Baddeley A (1990) Human memory: theory and practice. Erlbaum, Hillsdale NJ
- Bar-Hillel Y, Carnap R (1953) Semantic information. *Br J Philos Sci* 4:147–152
- Bartlett F (1932) Remembering. Cambridge University Press, Cambridge
- Bernal JD (1969) Science in history. Penguin Books, Harmondsworth
- Boden M (2016) AI. Oxford University Press, Oxford
- Bringsjord S, Govindarajulu NS (2020) Artificial Intelligence. In: Zalta EN (Ed.) The Stanford encyclopedia of philosophy (Summer 2020 ed.). <https://plato.stanford.edu/archives/sum2020/entries/artificial-intelligence/>
- Carstairs-McCarthy A (2001) Origins of language. In: Aronoff M, Rees-Miller J (eds) The handbook of linguistics. Basil Blackwell, Oxford
- Chandler D (2007) Semiotics: the basics. Routledge, London
- Chomsky N (1957) Syntactic structures. Mouton, The Hague
- Crystal D (1971) Linguistics. Penguin Books, Harmondsworth
- de Saussure F (1983) Course in general linguistics. Duckworth, London
- Derry TK, Williams TI (1960) A short history of technology. Dover, New York
- Dosse F (1992) *Geschichte des Strukturalismus* [The history of structuralism]. Fischer, Frankfurt Am Main
- Dreyfus H (1992) What computers (still) can't do? MIT Press, Cambridge
- Eco U (1976) A theory of semiotics. Indiana University Press, Bloomington
- Floridi L (2010) Ethics and information revolution. In: Floridi L (ed) Information and computer ethics. Cambridge University Press, Cambridge, pp 3–19
- Floridi L (2011) Philosophy of information. Oxford University Press, Oxford
- Floridi L (2013) The philosophy of information. Oxford univeristy press, Oxford
- Floridi L (2017) The logic of design as a conceptual logic of information. *Mind Mach* 27(3):495–519
- Floridi L, Taddeo M (2004) The symbol grounding problem: a critical review of a fifteen years of research. *J Exp Theor Artif Intell* 17:419–445
- Fodor J (1990) A theory of contents. MIT Press, Cambridge
- Ford M (2014) The rise of robots. Basic Books, New York
- Foucault M (1972) The Archeology of knowledge. Tavistock, London
- Franzen T (2005) Gödel's theorem. Peters, Wesley
- Fukuda K (2020) Science, technology and innovation ecosystem transformation toward society 5.0. *Int J Prod Econ* 220:2–14
- Garman M (1990) Psycholinguistics. Cambridge University Press, Cambridge
- Gödel K (1962) On formally undecidable propositions. Basic Books, New York
- Gunkel D (2012) The machine question. MIT Press, Cambridge
- Habermas J (1981) Theorie des kommunikativen Handelns 1–2 [Theory of communicative behavior]. Surkamp, Frankfurt am Main
- Habermas J (2009) Diskursethik [Discourse ethics]. Surkamp, Frankfurt am Main
- Harnad S (1990) The symbol grounding problem. *Phys D* 42(1–3):335–346
- Hawkes T (1977) Structuralism and semiotics. Routledge, London
- Hintikka J (1963) Knowledge and belief. Cornell University Press, Ithaca
- Hintikka J (1973) Logic, language-games and information. Clarendon Press, Oxford
- Hintikka J, Sandu G (1997) Game theoretical semantics. In: van Benthem J, Meulen A (eds) Handbook of logic and language. Elsevier, Amsterdam
- Hinton G (1981) Implementing semantic networks in parallel hardware. In: Hinton G, Anderson J (eds) Parallel models of associative memory. Erlbaum, Hillsdale
- Hodges A (2014) Alan Turing: the enigma. Princeton University Press, Princeton
- Johnson-Laird P (1983) Mental models: towards a cognitive science of language, inference, and consciousness. Harvard University Press, Cambridge
- Johnson-Laird P (2008) How we reason? Oxford University Press, Oxford
- Jones D, Snider C, Nassehi A, Yon J, Hicks B (2020) Characterising the Digital Twin: a systematic literature review. *CIRP J Manuf Sci Technol* 29:36–52
- Kåhre J (2002) The mathematical theory of information. Kluwer, Boston
- Kant I (1976) Kritik der reinen Vernunft. [The critique of pure reason]. Felix Meiner, Hamburg (**Original work published 1781**)

- Kleene S (1971) Introduction to metamathematics. Wolters-Noorhoff, Groningen
- Kohonen T (1977) Associative memory. Springer, Berlin
- Kuehne U (2005) Die Methode des Gedankenexperimenten. Surkamp, Frankfurt am Main
- Kuhn T (1962) The structure of scientific revolutions. University of Chicago Press, Chicago
- Leech GN (1983) Semantics. Penguin Books, Harmondsworth
- Lindsay P, Norman D (1977) Human information processing. Academic Press, New York
- Lucas JR (1964) Minds, machines and Gödel. In: Anderson AR (ed) Minds and machines. Prentice-Hall, Upper Saddle River, pp 43–59
- Lyons J (1977) Semantics 1–2. Cambridge University Press, Cambridge
- Lyons J (1995) Linguistic semantics: an introduction. Cambridge University Press, Cambridge, UK
- McCarthy RA, Warrington EK (1990) Cognitive neuropsychology. Academic Press, San Diego
- Minsky ML (1967) Computation: Finite and infinite machines. Prentice-Hall, Englewood Cliffs
- Myllylä M, Saariluoma P (2022) Expertise and becoming conscious of something. *New Ideas Psychol* 64:100916
- Newell A, Simon HA (1972) Human problem solving. Prentice-Hall, Englewood Cliffs
- Ogden C, Richards I (1923) The meaning of meaning. Routledge & Kegan Paul, London
- Passmore J (1957) A hundred years of philosophy. Penguin Books, Harmondsworth
- Peirce C (1940) Collected papers V-VII. Harvard University Press, Cambridge
- Petzold C (2008) The annotated Turing. Wiley, Indianapolis
- Raatikainen P (2021) Gödel's incompleteness theorems. In: Zalta EN (Ed.) The Stanford encyclopedia of philosophy (Spring 2021 ed.). <https://plato.stanford.edu/archives/spr2021/entries/goedel-incompleteness/>
- Rosen R (1991) Life itself: a comprehensive inquiry into the nature, origin, and fabrication of life. Columbia University Press, New York
- Russell S, Norvig P (1995) Artificial intelligence. Prentice-Hall, Upper Saddle River
- Saariluoma P (1997) Foundational analysis: presuppositions in experimental psychology. Routledge, London
- Saariluoma P, Rauterberg M (2016) Turing's error-revised. *Int J Philos Study* 4:22–41. <https://doi.org/10.14355/ijps.2016.04.004>
- Saariluoma P, Nevala K, Karvinen M (2006) Content-based analysis of modes in design engineering. In: Gero J, Goel A (eds) Design computing and cognition'06. Springer, Berlin, pp 325–344
- Saariluoma P, Cañas JJ, Leikas J (2016) Designing for life: a human perspective on technology development. Palgrave Macmillan UK, London
- Saariluoma P, Rousi R (2015) Symbolic interactions: towards a cognitive scientific theory of meaning in human technology interaction. *J Adv Humanit* 3(3): 310–324. <http://cirworld.org/journals/index.php/JAH/article/view/100na>
- Saariluoma P, Cañas J, Karvonen A (2021) Human digital twins and cognitive mimetic. In: Human Interaction, Emerging Technologies and Future Applications III: Proceedings of the 3rd International Conference on Human Interaction and Emerging Technologies: Future Applications (IHET 2020), August 27–29, 2020, Paris, France (pp. 97–102). Springer International Publishing
- Salomaa A (1985) Computation and automata. Cambridge University Press, Cambridge
- Searle J (1991) Mind, brains & science. Penguin Books, London
- Shannon C (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423 (**623–656**)
- Shannon C, Weaver W (1949) The mathematical theory of communication. Illinois University Press, Urbana
- Tegmark M (2017) Life 3.0. Penguin Books, St Ives
- Turing AM (1948) Intelligent machinery. In: Copeland J (ed) The essential Turing. Clarendon Press, Oxford
- Turing AM (1950) Computing machinery and intelligence. *Mind* 59:433–460
- Turing AM (1936–7) On computable numbers, with an application to the entscheidungsproblem. *Proc Lond Math Soc* 42: 230–265
- von Wright GH (1956) Logiikka, filosofian ja kieli [Logic, philosophy and language]. WSOY, Porvoo
- Waldrop MM (2018) The dream machine: JCR Licklider and the revolution that made computing personal, 4th edn. Stripe Press, San Francisco
- Watson J (1914) Behaviorism. Keagan Paul, London
- Wiener N (1948) Cybernetics. MIT Press, Cambridge
- Wittgenstein L (1953) Philosophical investigations. Basil Blackwell, Oxford
- Wittgenstein L (1961) Tractatus-Logico philosophicus. Routledge & Kegan Paul, London (**Original work published 1921**)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.