Proceedings of Machine Learning Research 159:28–48, 2021 International Workshop on Self-Supervised Learning

A Unified Model of Reasoning and Learning

Pei Wang pei.wang@temple.edu
Department of Computer and Information Sciences, Temple University
Philadelphia, PA 19122-1801, USA

Editor: Kristinn R. Thórisson and Paul Robertson

Abstract

This paper analyzes the historical development of the conceptions of "reasoning" and "learning", especially their separation in the study of artificial intelligence and the attempts to combine them in various ways. A unified treatment of cognitive functions is provided in the AGI model NARS, where reasoning and learning are different facets of the same underlying process.

Keywords: reasoning, learning, NARS, unification

1. Introduction

Reasoning and learning are important cognitive functions that have been part of artificial intelligence (AI) research from its early days (McCarthy et al., 1955; Feigenbaum and Feldman, 1963).

In AI, reasoning and learning have been studied separately (Luger, 2008; Russell and Norvig, 2010; Poole and Mackworth, 2017), though in many cases both are needed for the desired results. Currently most of these situations are handled in a case-by-case manner, where these two cognitive functions are carried out separately and combined according to the specific situation of the problem.

To achieve general-purpose AI—*artificial general intelligence* (AGI) (Wang and Goertzel, 2007)—these two cognitive functions, as well as many others, should be integrated in a domain-independent manner. Various cognitive architectures have been proposed with similar objectives (Duch et al., 2008; Kotseruba and Tsotsos, 2020).

In this article, a unified model of reasoning and learning (as well as other cognitive functions) is introduced. This model, NARS (Non-Axiomatic Reasoning System), has been designed according to a theory of intelligence as a whole, where various cognitive functions are developed to serve the overall objectives of the system (Wang, 1995, 2006, 2013).

In the following, I will start by reviewing the development of the notions of "reasoning" and "learning" to explain the historical root for their separation in AI research, then introduce the most relevant parts of NARS, and finally summarize the features of this approach using a concrete example.

2. Historical Development of the Notions

To fully understand the current issues, it is necessary to briefly trace the development of the notions of "reasoning" and "learning" in history.

2.1. How reasoning has become mainly about proofs

The study of reasoning started mostly in logic, which focuses on valid reasoning and normative models (Kneale and Kneale, 1962).

The following are some representative works in traditional logic:

- Aristotle's syllogism is a set of valid inference rules, to be followed in thinking and debating in general (Aristotle, 1989).
- Leibniz attempted to extend the traditional syllogism to a "universal calculus" so that differences in people's opinions can be resolved by calculation (Kneale and Kneale, 1962).
- Boole hoped to find the "the laws of thought" by treating logic as an algebra (Boole, 1854).

The overall objective of these works was to identify and formalize the patterns of valid reasoning (or call it inference) in human thinking in general, though the focused inference type was deduction, since its validity is relatively easy to justify. Other types of reasoning, such as induction and abduction, were considered to be cognitively useful, but not truth-preserving, as they may derive false conclusions from true premises (Hume, 1748; Peirce, 1931). The typical domain of application of logic is mathematics, where deduction takes the central role, though it was usually explicitly or implicitly assumed that reasoning in other domains, such as natural science and everyday life, should follow the same logic, at least approximately.

This mathematical orientation was made explicit and even exclusive by the works of Frege (1999) and Whitehead and Russell (1910), which have been properly referred to as "mathematical logic", as these logic systems were developed primarily to provide a solid foundation for mathematics, and therefore take theorem proving as the canonical form of reasoning. The anti-psychologism in logic has become so strong that the reality of human reasoning is mostly taken as irrelevant and misleading.

As mathematical logic has dominated the field, research on non-mathematical logic has been under the rubric of "philosophical logic" (Grayling, 2001) where various types of non-classical logic are explored (Haack, 1996). A non-classical logic usually attempts to extend or modify classical logic to explain or reproduce some phenomenon or function observed in human reasoning, while largely keep the traditional framework and principles of classical logic as much as possible.

Though logicians have ignored psychological observations, psychologists have widely taken classical logic as "the logic" against which the rationality of human reasoning is judged. For instance, deviation from classical logic, as shown in Wason Selection Task, is judged as human fallacy (Wason and Johnson-Laird, 1972). Even competing psychological theories "mental logic" (Braine and O'Brien, 1998) and "mental models" (Johnson-Laird, 1983) are largely based on the syntax and semantics of classical logic, respectively.

While the study of individual inference steps is dominated by mathematical logic, the study of multi-step inference processes has been mostly guided by the notion of *effective procedure*. In mathematics, a problem-solving procedure is considered as "effective" if it consists of a finite number of exact, finite instructions, and for each problem instance the solving process is deterministic and always terminates. In theoretical computer science, this notion is formalized as "computation" in a Turing Machine (Hopcroft et al., 2007). For an axiomatic system, it corresponds to a decision procedure that judges whether an arbitrary proposition is a theorem.

With such an intellectual heritage, in AI the works on reasoning started in theorem proving, also known as "automated reasoning", which gradually develops into a domain of its own (Feigenbaum and Feldman, 1963; Robinson and Voronkov, 2001).

The cognitive functionality of reasoning is clearly not restricted to mathematics, and it is natural to treat reliable knowledge as axioms to derive their implications, which should also be reliable knowledge. Such approaches have been proposed to cover "naïve physics" (Hayes, 1979), expert knowledge (Buchanan and Shortliffe, 1985), and encyclopedic knowledge (Lenat, 1995).

However, reasoning outside mathematics, especially with "commonsense" knowledge, does not have the "from truth to truth" nature of theorem proving, so there is a need for new logic systems in which the validity of reasoning can be relaxed in certain ways. Major attempts to revise and extend the reasoning frameworks include the following:

- To open the reasoning system to new evidence, which may reject the tentative conclusions derived by default rules, or to revise the system's beliefs (McCarthy, 1989; Reiter, 1980; Alchourrón et al., 1985).

- To reason under uncertainty with numerical measurements according to probability theory or fuzzy logic (Nilsson, 1986; Pearl, 1988; Zadeh, 1983; Dubois and Prade, 2003)).

- To reason on procedural knowledge, so the conclusions can be executable, as in robot control (Fikes and Nilsson, 1971), logic programming (Kowalski, 1979), and agent systems (Rao and Georgeff, 1995).

Even after the above extensions and revisions, the study of reasoning in AI is still focused on variants of deduction and theorem proving, where the correctness of the conclusions is guaranteed by the correctness of the given knowledge, plus the validity of the inference rules. Though such techniques are useful, they are far from reaching what human reasoning achieves.

2.2. How learning has become mainly about algorithms

Learning in general includes various types of experience-driven changes in capacity or behavior (Reisberg, 1999). The scientific study of learning started in psychology and neuroscience, as exemplified by the works on Pavlovian conditioning (Rescorla and Wagner, 1972) and Hebbian learning (Hebb, 1949).

Since learning is widely recognized as a central component or aspect of intelligence, it was taken as an important topic in AI research from the field's very beginning (Turing, 1950; McCarthy et al., 1955; Samuel, 1959; Minsky, 1961). In the 1980's, "machine learning" became a field of its own (Michalski et al., 1984), with a diverse collection of approaches (Carbonell, 1989), including decision tree (Quinlan, 1986), genetic algorithm (Holland, 1986), and so on. Various types of artificial neuronal network (ANN) models are designed with learning as its central capability (Rosenblatt, 1957; Rumelhart and McClelland, 1986; LeCun et al., 2015; Schmidhuber, 2015), and after decades of explorations ANN has become the most remarkable achievement of machine learning.

Since the intuitive sense of "intelligence" is closely related to "problem-solving capability", it is quite common for the latter to be used to define or measure the former. Following this path, "learning" is widely considered as the increasing of a system's problem-solving capability. As in computer science "problem solving" is normally formalized as "computation in a Turing Machine", or equivalently, as "following an algorithm" (Cormen et al., 2009), learning corresponds to "algorithm improvement". Accurately, machine learning is often defined as a *meta-level computation* that uses a learning algorithm with training data as input, and produces a "model" for an object-level (practical) problem (Flach, 2012). As the model is also an algorithm that finds solutions for the instances of the practical problem, the learning process can be called "algorithmic" as it follows a (meta-level) algorithm to generate an (object-level) algorithm (Wang and Li, 2016).

The above conceptual analysis explains many features and limitations of the current mainstream machine learning techniques, in spite of their differences in details. For instance, learning systems usually have clearly separated training phase (which follows the *learning* algorithm) and working phase (which follows the *learned* algorithm), and therefore need special arrangements to learn during the working process. On the contrary, there is no such a clear separation in human cognition, where "learning" and "working" are relatively distinguished, and usually interweave with each other.

Since the result of learning is to get a model or function (input-output mapping), the learning process can be more accurately described as "function approximation" (as in supervised learning, where the objective is to generalize the input-output pairs in the training data) or "function optimization" (as in unsupervised learning and reinforcement learning, where the objective is to maximize the quality of a classification or a policy, also according to the training data). In both cases, the most common technique is to tune a parameterized function that has universal approximation capability, such as certain ANNs.

When solving specific problems, *computational* systems depend on human-designed algorithms, while *learning* systems only demand proper training data to generate the algorithms needed. Consequently, the scope of applicability of computer has been greatly increased by the progress in machine learning. However, when compared to human learning, algorithmic learning only captures certain special cases.

The above conclusion is supported by the collection of new concepts appearing in the field in recent years: *one-shot learning, multi-task learning, transfer learning, online learning, life-long learning, active learning, cumulative learning, semi-supervised learning, self-supervised learning,* and so on. What is shown by these concepts is that the features mentioned in them are not naturally provided by algorithmic learning, so have to be added with special effort. On the contrary, these features are all intrinsic in human learning altogether.

2.3. The relation of the two functions in AI

Given their different origins and conceptions, reasoning and learning had been taken to be two separate cognitive functions even before the field of AI was formed. Partly for this reason, in AI they have been largely studied independent of each other. This situation is often justified from a problem-solving perspective: to solve a specific problem, one of them is often enough (Dietterich, 2003; Sutton, 2019; Sejnowski, 2020).

However, there are clearly many situations where both functions are needed. To use them together, there are the following alternatives:

Hybrid: To combine the existing techniques of both domains into one system.

Integrated: To design a system with a reasoning module and a learning module that can work together.

Unified: To use a single technique for both functions.

Hybrid systems work well for special applications, where multiple techniques can be aligned in task-specific ways, such as in IBM Watson (Ferrucci et al., 2013) and the projects of my own team (Hammer et al., 2021). However, it is not easy to get a system that works in a wide range of situations where the techniques have to cooperate under various conditions, and especially unanticipated ones. There are recent claims that the way to go for AI is to combine deep learning with symbolic reasoning (Marcus, 2020), though how to make the two techniques compatible is still an open problem.

Integrated systems often take the form of a cognitive architecture (Newell, 1990; Sun, 1995; Anderson and Lebiere, 1998; Franklin, 2007; Chong et al., 2007). This approach is different from the hybrid approach, as the modules are designed together. A typical way is to use reasoning for routine problem solving, while to use learning for increasing the system's capability. A concrete example is Soar, where the routine works are carried out by production rules (as a variant of reasoning), while learning happens as "chunking" that generates new rules (Newell, 1990).

More recent attempts of integration happens between symbolic reasoning and connectionist learning, as in "neural-symbolic computing" (Garcez et al., 2019). Such approaches have the potential of combining the advantages of these two competing paradigms, though the largest issue is still the conceptual conflict of the two. Even though the human brain/mind complex can be described both at the conceptual level (as in psychology) and at the neural level (as in neuroscience), it cannot be meaningfully seen as an integration of a "conceptual part" and a "neural part".

Contrary to the hybrid and integrated approaches, a unified approach attempts to mainly depend on a single technique for both reasoning and learning. Therefore, conceptual consistency will not be an issue, but the challenge is to provide the functionalities.

Previous examples of using reasoning to carry out learning include the Inference Theory of Learning, where non-deductive reasoning, such as induction and analogy, are effectively learning rules (Michalski, 1993). Also, since Bayesian theorem can be interpreted either as a reasoning rule or a learning rule (Pearl, 1990; Heckerman, 1999), the two functions are naturally unified in a Bayesian network.

In recent years, the successes of deep learning drove many researcher to use this technique for reasoning. Here the basic idea is to see reasoning as a special type of input-output mapping, which can be learned just like other mappings (Santoro et al., 2017; Saxton et al., 2019; Banino et al., 2020; Minervini et al., 2020).

As usual, each of these approaches has its strength and weakness. This article is not a survey on this topic, though some of the issues will be further analyzed in the following.

3. Reasoning and Learning in NARS

This section briefly introduces how NARS carries out reasoning and learning as the same process. As NARS is a complicated system and has been covered in many publications, including (Wang, 1995, 2006, 2013), the papers at the author's website, and the OpenNARS project website[1], here only the most relevant aspects of NARS are described.

3.1. NARS as an AGI

NARS is an AI model aimed at capturing the essence of intelligence and realizing the major cognitive functions observed in the human mind. The model is based on the theory that *intelligence is the ability of adaptation under insufficient knowledge and resources* (Wang, 1995, 2019b).

Here "adaptation" is basically what "learning" means in psychology, though not the *algorithmic learning* specified previously.

"Insufficient knowledge" means the problems faced by the system are often novel, so the system has no existing algorithm to follow as in Turing computation, nor the training data required by the conventional learning algorithms to learn a problem-specific algorithm. Therefore, a different type of problem solving mechanism is needed.

The restriction of "insufficient resources" requires the system to work in *real time*, in the sense that new problems may appear at any moment, and usually come with various time requirements, such as to be finished as soon as possible. Since the demand and supply of resources—especially computational time—change from time to time, the system cannot depend on problem-specific algorithms, as their time expense is fixed for a given problem instance in a specific implementation, and has no flexibility.

Consequently, the working definition of intelligence accepted in NARS prevents the system from depending on Turing computation or algorithmic learning for each problem *class*, but have to directly process each problem *instance* in a case-by-case manner, using whatever knowledge and resource available.

NARS still consists of basic processes following predetermined algorithms, though not at the problem-solving or task-processing level. The granularity of its executable process is much smaller than the ordinary AI systems, as the "Peewee Granularity" suggested by Thórisson and Nivel (2009). The basic steps in NARS are organized flexibly at run time to solve various problems, similar to production rules (Newell, 1990) and codelets (Hofstadter and Mitchell, 1994). A key difference between NARS and the previous techniques is that as NARS is designed to be a general-purpose AI, or Artificial General Intelligence (Goertzel and Pennachin, 2007), it cannot be equipped with problem-specific steps, nor to leave that for the user to provide for each problem. Instead, the system needs a set of domain-independent processing steps that can be combined to solve a wide range (if not all) of problems.

Though the above request looks hard to meet at first glance, it is exactly what is expected from a *logic*, in the original and ordinary sense of the word, that is, a set of well-justified rules that each only takes a small amount of knowledge and resources to work, and can be combined to solve problems in many domains.

1. http://opennars.org/

This is why NARS is built as a reasoning system that follows a logic. Of course, it cannot be the proof-oriented mathematical logic or its close variants. Among the functions missed in ordinary reasoning systems, it is learning that NARS must have.

3.2. Extended form of reasoning

An adaptive reasoning system is fundamentally different from an axiomatic reasoning system, as the former must face a changing environment, so it is absolutely necessary for the system to be able to revise its beliefs according to its experience. With the assumption of insufficient knowledge, it means no belief can be taken as an "axiom" in the sense that its truth-value cannot be challenged by new experience, and this is why NARS has "non-axiomatic" in its name, which stresses its key difference from the traditional systems. On the other hand, NARS also has "reasoning system" in its name, as it still follows a logic.

Without a set of axioms as reference, how can NARS decide the truthfulness of a statement? Being adaptive, NARS uses an *experience-grounded semantics*, and judges the truth-value of each statement by its extent of agreement with the system's relevant experience, or available evidence (Wang, 2005). According to this definition, non-deductive reasoning—such as induction, abduction, analogy, and so on—become justifiable, as they are still truth-preserving in the sense that their premises support the conclusion to the extent indicated by the truth-value (Wang, 2013). Similarly, the meaning of a term used in the system is determined by its role in the experience, so may change as the system is running and getting new experience.

NARS' primary work is not theorem proving, but achieving its goals according to its beliefs obtained from its experience. Since accurately predicting the future is impossible, what reasoning does is to relate the current situation to the past situations to treat novel objects as familiar ones. In a broad sense, all reasoning is analogy as argued by Hofstadter (1995). For this reason, the basic statement in NARS does not describe the relation among objects in the world, but the substitutability among concepts, that is, to what extent concept A can be treated as concept B.

The formal language suitable for such statements is not predicate calculus, but the categorical language used in term logic, as exemplified by Aristotle's Syllogistic (Aristotle, 1989), where the typical form of a statement consists of a *subject term* and a *predicate term*, with a *copula* indicating their substitutability. Using these categorical statements as premises and conclusions, the syllogistic rules specify the various ways the substitutability transfer among the terms. In this format, Peirce found an elegant way to uniform deduction, induction, and abduction, by taking the latter two as the former one with a premise switched with the conclusion (Peirce, 1931).

In NARS, the rules used by Aristotle and Peirce are extended from binary to multi-valued, with the truth-value interpreted as a measurement of evidential support, so as to consistently justify deduction, induction, and abduction, as well as many other inference rules (Wang, 2013):

- The revision rule combines truth-values coming from disjoint evidence bases.
- The choice rule selects the best answer for a question by balancing evidential support, simplicity, and other factors.

- The compositional rules build compound terms from existing terms to express experience more efficiently.
- Statements are extended to represent events with time-stamped truth-value, so as to support temporal and causal reasoning.
- Events are extended to represent operations (executable/realizable statements) and goals (statements to be realized) to support procedural reasoning, as in logic programming (Kowalski, 1979).
- Via a sensorimotor interface, NARS can send commands and receive feedback from connected hardware/software devices, so as to control sensors and actuators.

Each problem usually takes multiple inference steps to solve. Since each rule is justified independently, there are many different ways to combine them for a given problem. Under the knowledge restriction, NARS normally does not know the optimal procedure (that is, the problem-specific algorithm); under resource restriction, the system cannot exhaustively explore every possible path.

In this situation, what NARS does is *controlled concurrency* (Wang, 2006), which is similar to the time-sharing mechanism in operating systems. Conceptually, there are many tasks under processing in parallel, and the system moves among them and carries out one inference step on one of them in each time slot. The system does not equally treat every task, or every possible path (formed by the beliefs accessed) of processing a task, but gives each of them a relative priority value to indicate its rank in resource allocation. Each priority value is a summary of the relevant factors evaluated according to the system's experience. Tasks and beliefs with higher priority have higher chance to be accessed, and priority values are adjusted according to the changes in the situation.

Consequently, the reasoning process for a given problem is not designed or learned in advance, but formed at run time when the problem instance is under processing. The processing path depends on the available knowledge and resources at the moment, so is highly context-sensitive. Even if the same problem instance is repeatedly given to the system, the process and the result may (though not necessarily) be different.

3.3. Learning as self-organizing

The architecture of NARS is shown in Figure 1. The system interacts with its environment via multiple channels. In some channels, input/output messages are in a language, including Narsese (the native language of NARS), a computer language, or a human language. Some other channels connect sensorimotor devices that directly interact with the physical world. Each channel manages the input and output operations, converts the data formats, and carries out some preliminary processing.

All input messages are treated as reasoning tasks by the system. There are three types of task: a *judgment* to be remembered, a *goal* to be achieved, or a *question* to be answered. Some (selected) tasks from the channels are pooled in the overall experience buffer, where cross-modality relations are built among them, and some (selected) tasks are entered into the system's memory for further processing.

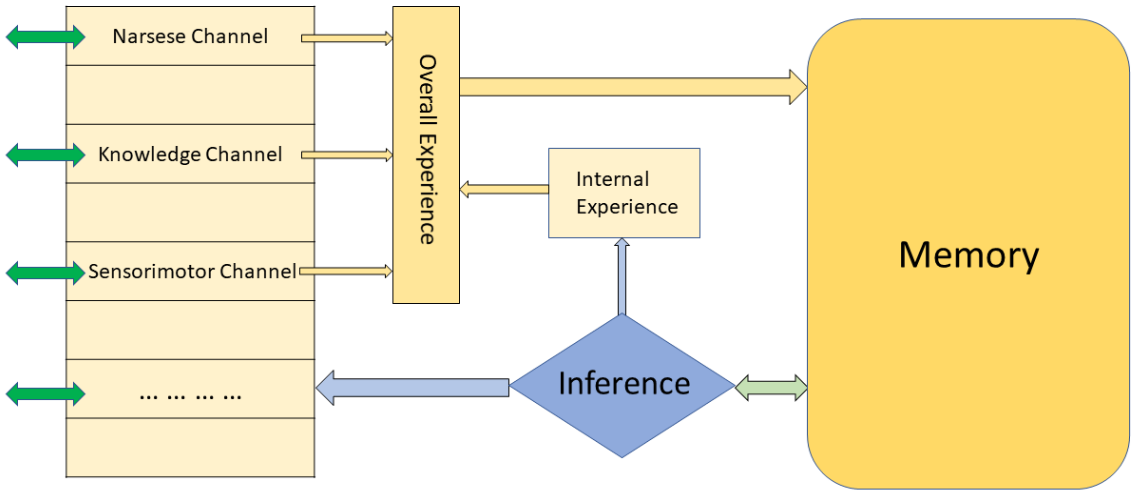

Figure 1: Architecture of NARS.

NARS' memory is a concept network, where each concept is named by a term, and contains the beliefs, desires, and tasks about that term. In each working cycle, a concept is selected, then a task and a belief (or desire) are select, and all these selections are biased by the priority of the items involved.

The selected task-belief pair is feed into the inference engine, where they are used as premises by the applicable rules and produce a number of derived tasks. The derived tasks are collected in the internal experience buffer for preliminary processing, and the selected ones among them enter the overall experience buffer, just like the selected input tasks.

After starting, NARS simply repeats the above working cycle until it is stopped from the outside. In the process, each task is processed using the system's knowledge (i.e., beliefs and desires), and at the same time enrich or revise the knowledge. The system's objective is not to process any specific task to a predetermined ending condition, but to carry out all existing tasks as far as possible, that is, as allowed by the available knowledge and resources.

The above is only a highly simplified description of how NARS works, and more details of the systems can be found in the relevant publications like Wang (2006, 2013), plus the online documentation and source code of the implementation. For the purpose of this article, it is enough to show the pervading nature of learning in NARS, as it happens in various forms in many places in the system:

- New inputs from the channels form the system's experience, from which the system learns various forms of knowledge at different levels of generalization and abstraction.
- The judgment tasks are processed in spontaneous forward inference that creates new beliefs and revise previous beliefs.
- The goals and questions are processed in backward inference that creates new desires and revise previous desires.

- Compound terms are composed to summarize experience, including new percepts and operations as special cases. Compound operations work like problem-specific skills or programs.
- The priority distributions among tasks, beliefs, desires, and concepts are adjusted from time to time according to history and context.
- New concepts are formed both from new compound terms and from significantly changed old concepts.

Overall, these experience-driven changes reorganize the system's memory to better predict the future and to use the computational resources more efficiently, so they are indeed "learning" in the original and general sense of the word (Wang, 2000; Wang and Li, 2016). However, it is very different from the current machine learning techniques:

- There is no overall learning algorithm.
- The system accepts multiple types of knowledge or data, rather than only the concrete problem instances.
- The learning process works with any amount of data, and no converging is expected.
- The system learns in real time and is sensitive to contextual time pressure.
- The reasoning processes and conclusions are explainable, at least in principle.

Now we can see why it is claimed at the beginning of the article that reasoning and learning are unified in NARS, as there is no separate processes for each of the two. Even so, it is still meaningful to talk about the two functions separately: when the process is considered as reasoning, the focus is at the individual steps, with its premise-conclusion relations and their computational implementation; when the process is considered as learning, the focus is at the long-term consequences of the steps in memory.

4. An Example

In this section, a concrete example is used to show how NARS reasons and learns. Limited by the length of the article, the description is highly simplified to keep the basics, and only includes a small portion of the cognitive functions of NARS.

Representation

In this example, only the simplest Narsese statement is used, which has the format of "$S \rightarrow P$", where S is the subject term, P the predicate term, and '$\rightarrow$' the *inheritance* copula. The statement intuitively means "S is a specialization of P", or equivalently, "P is a generalization of S". To make the examples understandable, English words are used for terms, though in NARS the meaning of a term is determined by what the system knows about it, which will not be the same as what the same word means to a human reader, though there are certain similarity between the two.

The truth-value of a statement is written as $\langle f, c \rangle$, a pair of real numbers in [0, 1], where f is *frequency*, measuring the proportion of positive evidence among current evidence, and c is *confidence*, measuring the proportion of current evidence among all evidence at a near future after a constant amount of new evidence arrives. The formal definition of *evidence* in NARS is given in (Wang, 2013), and it is enough to be intuitively understood in this article. To make the description simple, in the following all given "facts" have the same truth-value $\langle 1.0, 0.9 \rangle$, that is, as supported by the same amount of positive evidence, and no negative evidence.

This example starts with three facts:

$$bird \rightarrow animal \ \langle 1.0, 0.9 \rangle \tag{1}$$

$$robin \rightarrow bird \ \langle 1.0, 0.9 \rangle \tag{2}$$

$$\{Tweety\} \rightarrow robin \ \langle 1.0, 0.9 \rangle \tag{3}$$

They intuitively express "Bird is a type of animal", "Robin is a type of bird", and "Tweety is a robin", respectively. Curly braces are used in $\{Tweety\}$, because Tweety is a proper name, so cannot be treated as "a type of robin" until the *extensional set* operator (written as curly braces) turns Tweety into "Tweety-like things", intuitively speaking.

Deduction

As a term logic, typical inference rules in NARS are syllogistic, in that each rule takes two premises sharing a term, and derives a conclusion between the other two terms. Since statements in NARS have numerical truth-values, each inference rule has an associated truth-value function to calculate the truth-value of the conclusion from those of the premises.

The two factors in a truth-value are interpreted as extended Boolean variables, and the Boolean operators AND, OR, and NOT are extended from {0, 1} to [0, 1], similar to how Triangular Norms are used (Bonissone, 1987). Using these operators as building blocks, the truth-value functions of NARS are established from their boundary conditions determined according to the experience-grounded semantics. Since truth-value calculation is not the focus of this article, in the following the results are directly displayed and their properties are discussed, without explaining the truth-value functions involved. For the formal definitions and derivations of all the truth-value functions of NARS, see (Wang, 2013).

The deduction rule requires the subject of the first premise to be the same as the predicate of the second premise. Using pairs (1)-(2) and (2)-(3) as premises, the deduction rules derived (4) and (5), respectively:

$$robin \rightarrow animal \ \langle 1.0, 0.81 \rangle \tag{4}$$

$$\{Tweety\} \rightarrow bird \ \langle 1.0, 0.81 \rangle \tag{5}$$

Finally, from (4)-(3) or (1)-(5), the following conclusion is derived:

$$\{Tweety\} \rightarrow animal \ \langle 1.0, 0.73 \rangle \tag{6}$$

In the above deductions, since all evidence involved are positive, the conclusions all have frequency 1.0 (purely positive), though the confidence values get lower and lower with the increasing of inference steps, indicating the decrease of the stability of the judgments when challenged by new evidence.

If the truth-values in the premises and conclusions are all omitted and all statements are taken as (binary) *true* statements, the inference remains valid. It shows that the deduction rule extends the transitivity of the inheritance copula from binary to multi-valued.

Induction

The induction rule requires the two premises to share the same subject. In this example, after the system is given

$$\{Tweety\} \to [yellow]\ \langle 1.0, 0.9\rangle \tag{7}$$

where square brackets are used in $[yellow]$, because *yellow* is an adjective, so needs to be turned into "yellow things" by the *intensional set* operator (written as square brackets), so as to serve as the predicate in an inheritance statement.

From (3) and (7), the induction rule derives the following conclusion

$$robin \to [yellow]\ \langle 1.0, 0.45\rangle \tag{8}$$

What happened here is the property of "being yellow" is generalized from $\{Tweety\}$ to *robin*. Since the conclusion states about a larger number of situations than the premises, such an inference is "ampliative" and therefore invalid according to the traditional theories, as argued by Hume (1748). Induction and other non-deductive inference become justifiable in NARS, because reasoning is no longer taken as theorem proving but a form of adaptation, or learning, where a conclusion is evaluated against *past experience*, not *future experience* or *objective reality*. In the above example, $\{Tweety\}$ provides a piece of positive evidence for the conclusion, as indicated by the truth-value. Again, the detail of truth-value calculation is explained in (Wang, 2013), though it can still be seen that with the same truth-values of the premises, inductive conclusions are less confident than deductive conclusions. In NARS, induction is a form of "weak" inference while deduction is "strong". Therefore, the traditional distinction between the two types of inference still exists in NARS, except that here it is a *quantitative* difference, not a *qualitative* one.

Given the symmetry of the premises, from (7) and (3) the induction rule also derives

$$[yellow] \to robin\ \langle 1.0, 0.45\rangle \tag{9}$$

because positive evidence supports inheritance in both directions. However, since negative evidence only effects one of the two conclusions, (8) and (9) may get different truth-values in the long run. This will become more clear after the revision rule is described in the following.

For the same reason, (5) and (6) can be used with (7) to get more inductive conclusions:

$$bird \to [yellow]\ \langle 1.0, 0.42\rangle \tag{10}$$

$$animal \to [yellow]\ \langle 1.0, 0.40\rangle \tag{11}$$

They show that NARS can generalize the same observation to different levels. Unlike in machine learning algorithms, in NARS there is no "inductive bias" that favors certain specific generalizations among all possibilities. Instead, different levels of generalizations can coexist, though usually with different truth-values and usages. Of course, NARS will not attempt to exhaust all possible generalizations of an observation, but only produces the ones obtained under the existing knowledge and resource restriction.

Abduction

The abduction rule is symmetric with the induction rule, and requires the two premises to share the same predicate. In this example, after the system is given

$$goldfinch \rightarrow [yellow] \ \langle 1.0, 0.9 \rangle \tag{12}$$

from (12) and (7) by abduction, it is derived

$$\{Tweety\} \rightarrow goldfinch \ \langle 1.0, 0.45 \rangle \tag{13}$$

In this case, "being yellow" provides positive evidence for Tweety to be judged as a goldfinch, though abduction is also a form of weak inference, so the confidence of the conclusion is relatively low, as in induction.

Peirce (1931) considered the cognitive function of abduction to be *explanation*, which can be applied to this example, that is, "Tweety is a goldfinch" explains why it is yellow. However, in NARS the rules are defined and applied in a formal way, so to say the above inference is abduction, it is completely because of the pattern it has.

Revision

In each inference step, only the premises are considered, so the above abductive step does not consider that Tweety cannot be both a robin and a goldfinch. However, further inference may reveal such contradictions, then the revision rule will be invoked. The premises of the revision rule are two judgments that are about the same statement, but their truth-values are based on disjoint bodies of evidence. In the conclusion, the evidence from both premises are pooled, and the truth-value is calculated accordingly.

To continue the current example, assume from some source the system gets the following judgment that has negative evidence only:

$$\{Tweety\} \rightarrow animal \ \langle 0.0, 0.9 \rangle \tag{14}$$

From (14) and the earlier conclusion (6), the revision rule generates

$$\{Tweety\} \rightarrow animal \ \langle 0.23, 0.92 \rangle \tag{15}$$

which is mostly negative, because (14) is based on more evidence than (6).

Since (6) is derived from other judgments, the revision process does not stop here. From (14) and the relevant judgments, eventually the following revision conclusions will be produced, where the step-by-step process is omitted:

$$robin \rightarrow animal \ \langle 0.84, 0.83 \rangle \tag{16}$$

$$\{Tweety\} \rightarrow bird \ \langle 0.84, 0.83 \rangle \tag{17}$$

$$bird \to animal \ \langle 0.93, 0.91 \rangle \tag{18}$$

$$\{Tweety\} \to robin \ \langle 0.93, 0.91 \rangle \tag{19}$$

$$robin \to bird \ \langle 0.96, 0.90 \rangle \tag{20}$$

This example shows that even the initial input "facts" can be revised to different extents, depending on their relations with other knowledge obtained from the experience.

Composition

From (7) and (5), a composition rule derives the conclusion

$$\{Tweety\} \to ([yellow] \cap bird) \ \langle 1.0, 0.73 \rangle \tag{21}$$

This rule is different from the other rules introduced earlier in that it constructs a compound term that intuitively means "yellow bird", which was not in the premises.

Composition rules build various types of compound terms from the terms in the premises, as attempts to find patterns and to summarize experience. This is one way to create new concepts in the system. It is usually impossible to decide whether a new concept will be valuable, as the future is not accurately predictable. What NARS does is to let concepts compete for resources, and gradually form a relatively stable concept network in memory with the concepts that have been useful in summarizing the system's experience and accomplishing its tasks, while the other concepts are forgotten, sooner or later.

The above description of the example focuses on the inference rules, though at the same time there are other activities going on, such as the attention allocation mechanism that selects the premises in each inference step. The selection is priority-based, but to describe that aspect of the system will be too complicated for this article.

Now we can see that each of the above inference steps can be considered as both reasoning and learning, according to certain interpretation of these notions. When the focus is on the relationship between the input and output of each step, we see reasoning; when the focus is on the consequence of each step, we see learning.

5. Conclusions

What distinguish NARS from the other AI/ML projects are primarily in its working definitions of the following basic concepts:

Problem solving. Turing computation properly captures the meaning of "problem solving" in mathematics and computer science, where a problem is defined on a set of instances, and the solving process should be repeatable and terminable. However, for adaptive systems, problem solving should be case-by-case, depending on the available knowledge and resources at the moment. The Turing computation definition is too restrictive, not because of the transition function, but the initial and final states, since the unique initial state excludes memory between computation processes, and the predetermined final states exclude context sensitivity in deciding the standard for solutions. What an adaptive system needs is *relative rationality* (Wang, 2011) and the corresponding notion of problem solving.

Intelligence. Though currently "intelligence" has multiple major understandings and each has its theoretical and practical values, there are reasons to define it as "adaptation with insufficient knowledge and resources", as such a definition gives intelligence a domain-independent identity by taking it as a meta-level capability. On the contrary, the current focus on problem-solving confuses *intelligence* with *skills*, and consequently fails to distinguish intelligent mechanisms from computational mechanisms (Wang, 2019b). NARS is a realization of this working definition of intelligence, and shows many advantages over the other approaches.

Reasoning. The type of reasoning captured by mathematical logic is only a special case of human reasoning, and there are still types of valid inference outside axiomatic systems. These two kinds of reasoning systems, axiomatic and non-axiomatic, with their corresponding logic, are suitable for different situations and purposes. What is needed for AI most is to recognize the patterns of valid reasoning in everyday thinking, then to formalize and automatize them, as exemplified by NARS (Wang, 2019a).

Learning. Though machine learning research has achieved great successes, what they have been focused on is only a special type of learning, when compared to the types of learning occurring in human cognition. In NARS, learning is taken as self-organizing activities that occur in many forms and many places in the system, and shows many desired features (Wang and Li, 2016).

In scientific research, usually we should use a concept with its generally accepted definition and understanding, unless there is enough reason to challenge the consensus. Here my major reason is that the above new definitions provide better solutions to many existing problems, and are also arguably closer to the original meanings of these concepts.

One consequence of the new conceptions is the natural unification of the various cognitive functions. In this article only the unification of reasoning and learning is discussed, though in the same spirit many other cognitive functions are also carried out by the same underlining process in NARS.

Given the complexity of intelligence, cognition, thinking, and the related notions, people often focus on one aspect of them in research, which is a valid and feasible strategy. However, just because we can recognize a cognitive function and describe it coherently, it does not mean that it should be realized in computers as a process independent of the other cognitive functions. On the contrary, it is very likely to be one aspect of a underlying process that is also responsible for many other cognitive functions altogether. In such a situation, to achieve them together may be not only theoretically more coherent, but also technically easier. NARS is still an on going project, and its further progress will hopefully tell us more about this strategy toward general intelligence.

Acknowledgments

Support for this research partly comes from the University Research Program of Cisco Systems, Inc., though the funding has no influence on the opinions expressed in this article. Thanks Kris Thórisson and the reviewers for their valuable comments.

References

C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

John R. Anderson and Christian Lebiere. *The Atomic Components of Thought.* Lawrence Erlbaum Associates, Mahwah, New Jersey, 1998.

Aristotle. *Prior Analytics.* Hackett Publishing Company, Indianapolis, Indiana, 1989. Translated by R. Smith.

Andrea Banino, Adri Puigdomnech Badia, Raphael Kster, Martin J. Chadwick, Vinicius Zambaldi, Demis Hassabis, Caswell Barry, Matthew Botvinick, Dharshan Kumaran, and Charles Blundell. Memo: A deep network for flexible combination of episodic memories. In *International Conference on Learning Representations*, 2020.

Piero P. Bonissone. Summarizing and propagating uncertain information with Triangular Norms. *International Journal of Approximate Reasoning*, 1:71–101, 1987.

George Boole. *An Investigation of the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities.* Walton and Maberly, London, 1854.

Martin D. S. Braine and David P. O'Brien, editors. *Mental Logic.* Lawrence Erlbaum Associates, Mahwah, New Jersey, 1998.

Bruce G. Buchanan and Edward H. Shortliffe, editors. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1985.

Jaime G. Carbonell. Introduction: Paradigms for machine learning. *Artificial Intelligence*, 40(1-3):1–9, 1989.

Hui-Qing Chong, Ah-Hwee Tan, and Gee-Wah Ng. Integrated cognitive architectures: a survey. *Artificial Intelligence Review*, 28(2):103–130, 2007.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 3rd edition, 2009.

Thomas Dietterich. Learning and reasoning. Technical report, School of Electrical Engineering and Computer Science, Oregon State University, 2003.

Didier Dubois and Henri Prade. Fuzzy set and possibility theory-based methods in artificial intelligence. *Artificial Intelligence*, 148:1–9, 2003.

Włodzisław Duch, Richard Oentaryo, and Michel Pasquier. Cognitive architectures: where do we go from here? In *Proceedings of the First Conference on Artificial General Intelligence*, pages 122–136, 2008.

Edward A. Feigenbaum and Julian Feldman, editors. *Computers and Thought.* McGraw-Hill, New York, 1963.

David Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T. Mueller. Watson: Beyond Jeopardy! *Artificial Intelligence*, 199:93–105, 2013.

Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.

Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York, 2012.

Stan Franklin. A foundational architecture for artificial general intelligence. In Ben Goertzel and Pei Wang, editors, *Advance of Artificial General Intelligence*, pages 36–54. IOS Press, Amsterdam, 2007.

Gottlob Frege. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. In Jean van Heijenoort, editor, *Frege and Gödel: Two Fundamental Texts in Mathematical Logic*, pages 1–82. iUniverse, Lincoln, Nebraska, 1999. Originally published in 1879.

Artur d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6(4):611–632, 2019.

Ben Goertzel and Cassio Pennachin, editors. *Artificial General Intelligence*. Springer, New York, 2007.

Anthony C. Grayling. *An Introduction to Philosophical Logic*. Wiley-Blackwell, Malden, MA, 3rd edition, 2001.

Susan Haack. *Deviant Logic, Fuzzy Logic: Beyond the Formalism*. University of Chicago Press, Chicago Press, 1996.

Patrick Hammer, Tony Lofthouse, Enzo Fenoglio, Hugo Latapie, and Pei Wang. A reasoning based model for anomaly detection in the Smart City domain. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Intelligent Systems and Applications*, pages 144–159, Cham, 2021. Springer International Publishing.

Patrick J. Hayes. The naïve physics manifesto. In Donald Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 242–270. Edinburgh University Press, Edinburgh, 1979.

Donald O. Hebb. *The Organization of Behavior*. Wiley & Sons, New York, 1949.

David Heckerman. Bayesian learning. In Robert A. Wilson and Frank C. Keil, editors, *The MIT Encyclopedia of the Cognitive Sciences*, pages 70–72. MIT Press, Cambridge, Massachusetts, 1999.

Douglas R. Hofstadter. On seeing A's and seeing As. *Stanford Humanities Review*, 4: 109–121, 1995.

Douglas R. Hofstadter and Melanie Mitchell. The Copycat project: a model of mental fluidity and analogy-making. In K. Holyoak and J. Barnden, editors, *Advances in Connectionist and Neural Computation Theory, Volume 2: Analogical Connections*, pages 31–112. Ablex Publishing Corporation, Norwood, New Jersey, 1994.

John H. Holland. Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning: an artificial intelligence approach*, volume II, pages 593–624. Morgan Kaufmann, Los Altos, California, 1986.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Boston, 3rd edition, 2007.

David Hume. *An Enquiry Concerning Human Understanding*. London, 1748.

Philip Johnson-Laird. *Mental Models*. Harvard University Press, Cambridge, Massachusetts, 1983.

William Kneale and Martha Kneale. *The development of logic*. Clarendon Press, Oxford, 1962.

Iuliia Kotseruba and John K. Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94, 2020.

Robert Kowalski. *Logic for Problem Solving*. North Holland, New York, 1979.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson, Boston, 6th edition, 2008.

Gary Marcus. The next decade in AI: Four steps towards robust artificial intelligence, 2020. arXiv:2002.06177.

John McCarthy. Artificial intelligence, logic and formalizing common sense. In Richmond H. Thomason, editor, *Philosophical Logic and Artificial Intelligence*, pages 161–190. Kluwer, Dordrecht, 1989.

John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, 1955. URL: http://www-formal.stanford.edu/jmc/history/dartmouth.pdf, accessed in February, 10, 2022.

R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach*. Springer-Verlag, 1984.

Ryszard S. Michalski. Inference theory of learning as a conceptual basis for multistrategy learning. *Machine Learning*, 11:111–151, 1993.

Pasquale Minervini, Matko Bonjak, Tim Rocktschel, Sebastian Riedel, and Edward Grefenstette. Differentiable reasoning on large knowledge bases and natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5182–5190, 2020.

Marvin Minsky. Steps towards artificial intelligence. *Proceedings of the Institute of Radio Engineers*, 49:8–30, 1961.

Allen Newell. *Unified Theories of Cognition.* Harvard University Press, Cambridge, Massachusetts, 1990.

Nils J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

Judea Pearl. Jeffrey's rule, passage of experience, and Neo-Bayesianism. In Henry E. Kyburg, R. P. Loui, and Carlson G. N., editors, *Knowledge Representation and Defeasible Reasoning*, pages 245–265. Kluwer Academic Publishers, Amsterdam, 1990.

Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce*, volume 2. Harvard University Press, Cambridge, Massachusetts, 1931.

David L. Poole and Alan K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents.* Cambridge University Press, Cambridge, 2 edition, 2017.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

Anand S. Rao and Michael P. Georgeff. BDI-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, 1995.

Daniel Reisberg. Learning. In Robert A. Wilson and Frank C. Keil, editors, *The MIT Encyclopedia of the Cognitive Sciences*, pages 460–461. MIT Press, Cambridge, Massachusetts, 1999.

Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

R.A. Rescorla and A.R. Wagner. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non reinforcement. In A.H. Black and W.F. Prokasy, editors, *Classical Conditioning II*, pages 64–99. Appleton-Century-Crofts, New York, 1972.

John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning.* MIT Press, Cambridge, Massachusetts, 2001.

Frank Rosenblatt. The perceptron: A perceiving and recognizing automaton. Report 85-60-1, Cornell Aeronautical Laboratory, 1957.

David E. Rumelhart and James L. McClelland. PDP models and general issues in cognitive science. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations*, pages 110–146. MIT Press, Cambridge, Massachusetts, 1986.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 3rd edition, 2010.

Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1):206–227, 1959.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4967–4976. Curran Associates, Inc., 2017.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020.

Ron Sun. Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, 75:241–295, 1995.

Rich Sutton. The bitter lesson, 2019. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html, accessed in February, 10, 2022.

Kristinn R. Thórisson and Eric Nivel. Achieving artificial general intelligence through peewee granularity. In *The Proceedings of the Second Conference on Artificial General Intelligence*, pages 222–223, 2009.

Alan M. Turing. Computing machinery and intelligence. *Mind*, LIX:433–460, 1950.

Pei Wang. *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence*. PhD thesis, Indiana University, 1995.

Pei Wang. The logic of learning. In *Working Notes of the AAAI workshop on New Research Problems for Machine Learning*, pages 37–40, Austin, Texas, 2000.

Pei Wang. Experience-grounded semantics: a theory for intelligent systems. *Cognitive Systems Research*, 6(4):282–302, 2005.

Pei Wang. *Rigid Flexibility: The Logic of Intelligence*. Springer, Dordrecht, 2006.

Pei Wang. The assumptions on knowledge and resources in models of rationality. *International Journal of Machine Consciousness*, 3(1):193–218, 2011.

Pei Wang. *Non-Axiomatic Logic: A Model of Intelligent Reasoning.* World Scientific, Singapore, 2013.

Pei Wang. Toward a logic of everyday reasoning. In Jordi Vallverdú and Vincent C. Müller, editors, *Blended Cognition: The Robotic Challenge*, pages 275–302. Springer International Publishing, Cham, 2019a.

Pei Wang. On defining artificial intelligence. *Journal of Artificial General Intelligence*, 10 (2):1–37, 2019b.

Pei Wang and Ben Goertzel. Introduction: Aspects of artificial general intelligence. In Ben Goertzel and Pei Wang, editors, *Advance of Artificial General Intelligence*, pages 1–16. IOS Press, Amsterdam, 2007.

Pei Wang and Xiang Li. Different conceptions of learning: Function approximation vs. self-organization. In Bas Steunebrink, Pei Wang, and Ben Goertzel, editors, *Proceedings of the Ninth Conference on Artificial General Intelligence*, pages 140–149, 2016.

P. C. Wason and P. N. Johnson-Laird. *Psychology of Reasoning: Structure and Content.* Harvard University Press, Cambridge, Massachusetts, 1972.

Alfred North Whitehead and Bertrand Russell. *Principia Mathematica.* Cambridge University Press, Cambridge, 1910.

Lotfi A. Zadeh. The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and System*, 11:199–227, 1983.