Structural Machines as Unconventional Knowledge Processors †

Rao Mikkilineni 1,* and Mark Burgin 2

1 Ageno School of Business, Golden Gate University, San Francisco, CA 94105, USA
2 Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095, USA; mburgin@math.ucla.edu
* Correspondence: rmikkilineni@ggu.edu
† Conference Morphological, Natural, Analog and Other Unconventional Forms of Computing for Cognition and Intelligence (MORCOM), Berkeley, CA, USA, 2–6 June 2019.

Published: 7 May 2020

Abstract: Knowledge systems often have very sophisticated structures depicting cognitive and structural entities. For instance, representation of knowledge in the form of a text involves the structure of this text. This structure is represented by a hypertext, which is networks consisting of linguistic objects, such as words, phrases and sentences, with diverse links connecting them. Current computational machines and automata such as Turing machines process information in the form of symbol sequences. Here we discuss based the methods of structural machines achieving higher flexibility and efficiency of information processing in comparison with regular models of computation. Being structurally universal abstract automata, structural machines allow working directly with knowledge structures formed by knowledge objects and connections between them.

Keywords: cognition; named set; knowledge structure; theory of oracles; structural machine

1. Introduction

Many philosophers, mathematicians and scientists have attempted to define knowledge and its relationship to information and its processing. Plato was perhaps the first to articulate that knowledge is always about something—some object or domain or mental concept [1]. In order to discern a knowledge object or domain, we have to name it. A name may be a label, number, idea, text, process, and even a physical object of a relevant nature. The named objects may be composed into knowledge structures which may have interobject and intraobject relationships and associated behaviors that may cause changes to their state, form or content. The role of information-processing structures is to discern the relationships and behaviors and evolve the state, form or content accordingly. Information in the strict sense includes a capacity to change structural elements in knowledge systems [2].

Knowledge systems exist in the world of structures, which belongs to the well-known existential triad of the world together with the physical world and the mental world. The knowledge objects or domains in the physical or mental world manifest themselves as structures. In the physical world, matter (containing physical, chemical and biological structures) contains energy. Similarly, structures contain information. Thus, “information is related to structures as energy is related to matter” [3]. Figure 1 shows the relationships between the physical world, the mental world, the digital world and the world of structures. Note that it is possible to treat the digital world as the mental world of digital devices—computers, computer networks and so on.

In the case of Turing machine implementation, the knowledge systems are represented by symbolic data structures and a central processor unit operates on them to process information. The
machine acts as a cognitive apparatus [4] with locality and the ability to form information processing structures where information flows from one apparatus to another with a velocity defined by the medium. Physical manifestation of information processing functions and provides structures.

Figure 1. The world of structures and digital information processing.

A physical framework to address fluctuations and nondeterministic interactions of these processes using the concepts of entropy, energy and phase transitions through reconfiguration of structures. Any algorithm that can be specified is transformed into an executable function using CPU and Memory. Functions operate on data structures representing domain knowledge and the computation evolves their current state to a new state. As long as there are enough resources (CPU and memory), the computation will continue as encoded in the algorithm. This is equivalent to a digital gene (representing well-specified executable process evolutions) assisting the execution of business processes. Cognition comes from the ability to encode knowledge structures and their processing to transform them from one state to another just as genes in biology do.

It is interesting to note that the Turing computable functions allow representation of algorithms that define neural networks which are used to model processes that are usually treated as nonalgorithmic, such as voice recognition, video processing, etc. Here, cognition comes from the ability to encode how to mimic neural networks in the brain modeling and processing information just as neurons in biology do. The digital neuron in Figure 1 executes cognitive processes that are not specified in the form of conventional algorithms. It also imitates biological systems collaborating with an apparatus that processes algorithms analogous to a gene where knowledge is encoded in DNA structures.

By its organization, a standard Turing machine processes words letter by letter in the sequential mode. Thus, to work with knowledge using the standard Turing machines, it is necessary in advance to present knowledge as linear structures. This essentially decreases efficiency of computation creating unnecessary barriers for knowledge processing. To increase efficiency, Turing machines with many tapes and with multidimensional tapes were introduced. They perform computations with more advanced data structures such as arrays and collections of arrays. Nevertheless, this also was not enough, and to improve efficiency and allow processing of not only symbols but also links between them, more advanced automata, such as Kolmogorov algorithms storage modification machines and relational machines were developed. However, all these relations define only structures of the first order, while knowledge structures can have much higher orders. Structural machines eliminate this restriction and further advance efficiency [5]. In the next section, we discuss knowledge processing by structural machines. Structural machines work with knowledge structures of arbitrary order transforming, not only elements of these structures or the content of these elements as conventional models of computation do, but also relations of different orders in the processed structures. This allows achieving higher flexibility and efficiency in comparison with regular models of computation, including both conventional and unconventional computing systems. Structural machines can also simulate such advanced computational automata such as Kolmogorov algorithms,
limit Turing machines, storage modification machines, relational machines and other models of computation. Being structurally universal abstract automata, structural machines work directly with knowledge structures, molecular and atomic structures, with structures studied and utilized in the topological quantum field theory (TQFT), and with structures of quantum information such as qubits.

2. Named Sets, Knowledge Structures, Structural Machines, the Theory of Oracles and Digital Information Processing Structures

Structural relationships exist between data, which are entities observed in the physical world or conceived in the mental world. These structures define the knowledge of them in terms of their properties such as attributes, relationships and dynamics of their interaction. Information processing structures organize evolution of knowledge structures by an overlay of cognitive knowledge structures, which model, monitor and manage the evolution of the information processing system.

The most fundamental structure is called a fundamental triad or a named set [6]. It has the following visual representation shown in Figure 2:

![Figure 2. Visual representation of a fundamental triad (named set).](image)

An entity or object with its own name is connected to another entity or object with its own name. The connection, which itself has a name, depicts the knowledge of the relationship and the behavioral evolution when a state change occurs in either object.

A knowledge structure [1,7] is composed of related fundamental triads, and any state change causes behavioral evolution based on the connections. The long and short of the theory of knowledge is that the attributes of objects in the form of data and the intrinsic, and ascribed knowledge of these objects in the form of algorithms and processes, make up the foundational blocks for information processing. Information processing structures utilize knowledge in the form of algorithms and processes that transform one state (determined by a set of data) of the object to another one with a specific intent. Information structures and their evolution using knowledge and data determine the flow of information. Living organisms have found a way not only to elaborate the knowledge of the physical objects, but also to create information processing structures that assist them in executing state changes.

The structural machine framework describes a process which allows information processing through transformation of knowledge structures. It involves a control device that configures and executes information processing operations on knowledge structures and manages the operations throughout its lifecycle using a processor. The processor uses the knowledge structures as input and delivers the processed information as knowledge structures in the output space.

In the special case where input knowledge structure and the output knowledge structures are words (symbols) and the process to be executed is an algorithm (a sequence of operations), then the structural machine becomes a Turing machine. The control is outside the Turing machine which provides the algorithm to be executed, assuring that the processor has the right resources to perform the operations and judge whether the computation is performed as expected. In essence, the functional requirements of the system that is under consideration such as business logic, sensor and actuator monitoring and control (the computed) etc., are specified as algorithms and are executed by the processor transforming the knowledge structures from the input space to the output space. Figure 3 shows the structural machine framework.
Proceedings 2020, 47, 26

Figure 3. Structural machine framework for information processing structures.

It is important to observe that when the controller is an operator, the processor is a stored program implementation of a Turing machine, the processing space is the memory, and the knowledge structures are symbolic data structures, we obtain the current state-of-the-art information-processing structure.

A knowledge structure composed of the named sets is shown in Figure 4.

Figure 4. A network of knowledge structures showing the state vectors.

The knowledge structure depicted here is a graph that captures both cognitive and structural domain knowledge. The picture captures hierarchical, decoupled and concurrent structures whose evolutionary behavior is captured by the fundamental triads or named sets. The knowledge systems go beyond the knowledge captured by taxonomies or ontologies emerging, by adding the connection and depicting the relationships and evolutionary behaviors.

The knowledge structures composed of named sets are processed using the structural machines. The processor space deciphers the connection relationships and behaviors to evolve the state. The controller allows the implementation of a cognizing agent overlay that manages the downstream processors and associated knowledge-structure evolution. The cognizing agents shown in Figure 5 are defined using the mathematical theory of oracles proposed and developed by Mark Burgin [8]. According to this theory, an agent system can work as an oracle, collecting information for the basic machines or for a more complex information processing devices, such as a network or computer cluster. For instance, in the theory of superrecursive algorithms and computation, an important type of oracle contains information on whether a Turing machine halts given definite input or not. As we know in this case, even simple inductive Turing machines are able to compute this and other information that is incomputable by Turing machines [9]. Consequently, it is possible to use inductive Turing machines as oracles for bringing information that is incomputable by Turing machines and other recursive algorithms information to various computing devices such as Turing machines or neural networks.
Figure 5. Oracles as cognizing agents managing the execution and evolution of knowledge structures.

The utilization of higher-order inductive Turing machines supplies Turing machines and other computing devices with even more powerful oracles, making the whole schema of oracle machines, for which the oracle Turing machine is only one particular case, a constructive device. When the agent that functions as an oracle collects information for the basic machine by search, it is a mining agent. When this agent obtains information for the basic machine by computation, it is a designing agent. The hierarchy of computational models determines a hierarchy of knowledge mining and designing agents. In this hierarchy, there are different types of agents with specific behavior. For instance, rigid agents transform only external information using the same program (algorithm) as the basic machine. At the same time, reflexive agents change their own program (algorithm) to achieve better results.

3. Conclusions

The introduction of structural machines has three objectives:

1. the elaboration of a more adequate model of contemporary computers, which work with and are directed by high-level programming languages;
2. the formation of mathematical tools for transition from computing with data to knowledge processing in the framework of artificial intelligence (AI);
3. the construction of a theoretical model for natural computation.

Structural machines can have many unit processors, which perform parallel and concurrent computations. There are many examples of structural machines.

1. Any individual can be treated as a structural machine with the brain as its processor and the functional space. Note that usually, people work in the inductive mode.
2. A group of people is a structural machine with several unit processors and the functional space, which comprises the brains of these individuals.
3. In essence, an advanced computer is also a structural machine.
4. A computer network, such as the World Wide Web, is a structural machine.
5. An individual with their computers is also a structural machine with several unit processors, which include utilized computers and the brain of the individual.
6. A cell phone is a structural machine.
7. An individual with their computers and cell phones is also a structural machine with several unit processors, which include utilized computers, cell phones and the brain of the individual.

Funding: This research received no external funding.

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

References

1. Burgin, M. Theory of Knowledge: Structures and Processes; World Scientific Books: Singapore, 2016.
2. Burgin, M. Information: Problems, Paradoxes, and Solutions. Triple C 2003, 1, 53–70.
3. Burgin, M. Information in the structure of the world. Int. J. Inf. Theor. Appl. 2011, 18, 16–32.
4. Mikkilineni, R.; Morana, G. Post-Turing Computing, Hierarchical Named Networks and a New Class of Edge Computing. In Proceedings of the 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 12–14 June 2019; pp. 82–87.

5. Burgin, M.; Adamatzky, A. Structural Machines as a Mathematical Model of Biological and Chemical Computers. Theory Appl. Math. Comput. Sci. 2017, 7, 1–30.

6. Burgin, M. Theory of Named Sets; Nova Science Publisher Inc.: New York, NY, USA, 2011.

7. Burgin, M. Data, Information and Knowledge. Information 2004, 7, 47–57.

8. Burgin, M. Inaccessible Information and the Mathematical Theory of Oracles. In Information Studies and the Quest for Transdisciplinarity: Unity through Diversity; World Scientific: New York, NY, USA; London, UK; Singapore, 2017; pp. 59–114.

9. Burgin, M. Superrecursive Algorithms; Springer: New York, NY, USA, 2005.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).