



Proceedings Going beyond Church–Turing Thesis Boundaries: Digital Genes, Digital Neurons and the Future of AI ⁺

Rao Mikkilineni

Ageno School of Business, Golden Gate University, San Francisco, CA 94105, USA; rmikkilineni@ggu.edu + Conference Theoretical Information Studies (TIS), Berkeley, CA, USA, 2–6 June 2019.

Published: 11 May 2020

Abstract: The Church–Turing thesis deals with computing functions that are described by a list of formal, mathematical rules or sequences of event-driven actions such as modeling, simulation, business workflows, etc. All algorithms that are Turing computable fall within the boundaries of the Church–Turing thesis. There are two paths to pushing the boundaries. The first is to address the limitation in the clause "ignoring resource limitations". The second is to search for computing models that solve problems that no ordinary Turing machine can solve using superrecursive algorithms. We argue that "structural machines" provide a new solution to managing both without disrupting the computation itself.

Keywords: sentience; artificial intelligence; knowledge structures; theory of oracles; structural machines

1. Introduction

As professor Burgin states [1,2], "Computers and other computing devices transform input data into output data. This situation is adequately reflected by mathematical models of automata, algorithms and computations, when input strings of symbols are transformed into output strings of symbols. data contain two types of information—information for material and abstract computing devices such as computers and Turing machines, and information for people. Information in data for computing devices together with computing hardware and software (programs) defines the process of computation when input data and data that are already in the device bring about an operation of the device, e.g., state transition. Information for people is explicated (observed and comprehended) by means of interpretation when raw data are interpreted and attributed becoming knowledge."

The ingenuity of von Neumann's stored program control implementation of the Turing machine [3,4] is that it provides a physical implementation of a cognitive apparatus to represent and transform data structures that are created by physical or mental worlds to represent the domain under consideration. Stored program control implementation of the Turing machine has allowed us to create general purpose computers and [4] "use them to deterministically model any physical system, of which they are not themselves a part to an arbitrary degree of accuracy." All algorithms that are Turing computable fall within the boundaries of the Church–Turing thesis [3,4], which states that "a function on the natural numbers is computable by a human being following an algorithm, ignoring resource limitations, if and only if it is computable by a Turing machine." Turing computable functions are those that are easily described by a list of formal, mathematical rules or a sequence of event-driven actions such as modeling, simulation, business workflows, interaction with devices, etc.

Figure 1 represents the implementation of Turing Machines as a cognitive apparatus 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.



Figure 1. The digital gene—a cognitive apparatus executing a process evolution specified by a function or algorithm.

Physical manifestation of information processing functions and structure with locality and velocity of information flow provides a framework of physics to address fluctuations and non-deterministic interactions of the structure using the concepts of entropy, energy and phase transitions through reconfiguration of structures.

Any algorithm that can be specified is made 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 without any regard to the state's past history (Markovian evolution). 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 evolution algorithms) 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 also include algorithms that define neural networks, which are used to model processes that cannot be described themselves as algorithms such as voice recognition, video processing, etc. Cognition here comes from the ability to encode how to mimic neural networks in the brain model and process information just as neurons in biology do. Figure 2 shows the digital neuron (executing the cognitive processes that cannot be specified in an algorithm) mimicking the biological systems.



Figure 2. Digital Neuron—a cognitive apparatus executing the processes that cannot be specified as a sequence of tasks but encoded in neural network algorithms.

Proceedings 2020, 47, 15

Turing machine implementations of information processing structures as Gödel [5] proved suffer from incompleteness and recursive self-reference and therefore require external agents to instruct them and judge their outputs [5,6]. Cockshott et al. [4] conclude their book "Computation and its Limits" with the sentence "Their logical limits arise when we try to get them to model a part of the world that includes themselves."

There are two new drivers that are testing the boundaries of the Church–Turing thesis:

- 1. The success of the general purpose computer has enabled current generation mobile, cloud and high-speed information processing structures whose main criterion for success of their computation is no longer their termination as Turing machines are designed [7], but their response to changes-their speed, generality and flexibility, adaptability and tolerance to error, faults and damage. Current business services demand non-stop operation and their performance adjusted in real-time to meet rapid fluctuations in service demand or available resources without interrupting service. The speed with which the quality of service has to be adjusted to meet the demand is becoming faster than the time it takes to orchestrate the myriad infrastructure components (such as virtual machine (VM) images, network plumbing, application configurations, middleware, etc.) distributed across multiple geographies and owned by different providers. It takes time and effort to reconfigure distributed plumbing coordinating with multiple suppliers, which results in increased cost and complexity. Any new architecture must eliminate moving VM images or reconfigure physical and virtual networks across distributed infrastructure and enable self-managing applications with or without VMs. In this paper we discuss a new class of applications that are managed by an overlay of cognizing agents that is aware of available resources, resources being used and the knowledge of processes to reconfigure the resources.
- 2. The general-purpose computers allow them to deterministically model any physical system, of which they are not themselves a part, to an arbitrary degree of accuracy. However, the model falls short in addressing non-deterministic fluctuations. While computerized sensors and actuators allow observations and deep learning algorithms using neural networks (digital genes) provide insights, actions require a knowledge of managing risk, and responding with appropriate action requires an ability to use history and past experience along with deep knowledge about the circumstance. Systems are continuously evolving and interacting with each other. Sentient systems (with the capacity to feel, perceive or experience) evolve using a non-Markovian process, where the conditional probability of a future state depends on not only the present state but also on its prior state history. Digital systems, to evolve to be sentient and mimic human intelligence, must include time dependence and history in their process dynamics. In this paper we discuss the role of knowledge structures and the structural machines to implement systems capable of managing deep knowledge, deep memory and deep reasoning to determine the process evolution based on history.

Church–Turing thesis boundaries are challenged when rapid non-deterministic fluctuations drive the demand for resource readjustment in real-time without interrupting the service transactions in progress. They are also challenged when making decisions to evolve the current state based on the past history and learning. The information processing structures must therefore become autonomous and predictive by extending their cognitive apparatus to include themselves and their behaviors from the past along with the information processing tasks at hand. In the next section, we describe a new approach that extends the digital genes and neurons to go beyond the Church–Turing thesis to implement self-managing software systems that are aware of their resource requirements, available resources, current usage and reconfigure themselves to manage the fluctuations in the demand for resources or their availability without interrupting their execution. In addition, we discuss a means to model domain knowledge in the form of knowledge structures, which incorporate various objects interacting with each other with well-defined relationships and interactive behaviors. The knowledge structure evolution depends on deep knowledge, deep memory and deep reasoning facilitated by cognizing agents.

2. Theory of Oracles and Digital Information Processing Structures Going beyond the Church-Turing Thesis

First, the new approach is aimed at addressing the resource limitations caused by nondeterministic fluctuations by organizing their evolution to model process dynamics under the supervision of intelligent agents who have the knowledge of both the computed and the available pool of computer resources. The computing hardware resources and software functions are arranged to optimally execute processes and evolve their dynamics using a network of digital genes and neurons described above where each end node can be either a gene or a neuron or subnetworks of executing a workflow (See Figure 3). The hardware resources are characterized by their parameters such as the required CPU, memory, network bandwidth, latency, storage throughput, IOPs and capacity. The efficiency of computation is determined by the required resources, while the expressiveness of the computational process dynamics is established by the information processing structure and connecting hardware units, such as servers or routers, along with their interactions within and with the external world.



Figure 3. Turing machine managed by cognizing agents.

An implementation of the new computing model is described [2], in detail and the theory of Oracles that describes and defines the cognizing agents is given in [8,9]. Cognizing agents determine the requirements for the application from a blueprint; have knowledge of available resources; and configure, monitor and reconfigure as required to assure non-stop computing processes with stateful auto-scaling, auto-failover and live migration of components in distributed cloud networks.

Second, the new computing model uses knowledge structures and structural machines to extend computations from the current sequential (concurrent and synchronous) model to concurrent, asynchronous and interactive systems, which have special requirements. Such distributed asynchronous systems must deal with issues of local consistencies versus global consistencies. The structural machines and the knowledge structures overcome this problem with a well-defined model of interaction among the components with entities (or actors), their relationships in the form of connections and behaviors that specify the evolution in the form of algorithms [1,9–11].

The knowledge structures are implemented as managed micro-services deployed in a docker container network orchestrated by Kubernetes.

3. Conclusions

The Turing computing model is extended in two ways, first by using the Oracle theory to implement cognizing agents that manage distributed computational resources to address the clause "ignoring resource limitations" in the Church–Turing thesis.

Second, the model is extended by using the knowledge structures and the structural machines to incorporate asynchronous and concurrent interacting distributed computing machines to address

global consistency using propagation of behaviors among interacting actors through message passing. The resulting hierarchical named microservice network using a Kubernetes provisioning stack provides all the cloud features such as elasticity, autoscaling, self-repair and live reconfiguration of information processing structures without reboot. The model is derived from a recent theoretical framework for unification of different models of computation using "structural machines." They are shown to simulate Turing machines and inductive Turing machines and are also proved to be more efficient than Turing machines. The structural machine framework with a hierarchy of controllers managing the named service connections provides dynamic reconfiguration of the service network from browser to database to address rapid fluctuations in the demand for or the availability of resources without having to reconfigure IP address-based networks.

Conflicts of Interest: The author declares no conflicts of interest.

References

- 1. Burgin, M. Theory of Knowledge: Structures and Processes; World Scientific Books: Singapore, 2016.
- 2. Burgin, M.; Mikkilineni, R. Cloud computing based on agent technology, super-recursive algorithms, and DNA, *Int. J. Grid Utility Comput.* **2018**, *9*, 193–204
- 3. Turing, A.M. The Essential Turing; Copeland, B.J., Ed.; Oxford University Press: Oxford, UK, 2004.
- 4. Cockshott, P.; MacKenzie, L.M.; Michaelson, G. *Computation and Its Limits*; Oxford University Press: Oxford, UK, 2012.
- 5. Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik* **1931**, *38*, 173–198.
- 6. Turing, A.M. Systems of logics based on ordinals. Proc. Lond. Math. Soc. 1939, 45, 161–228.
- 7. Dodig-Crnkovic, G.; Giovagnoli, R. Computing Nature—A Network of Networks of Concurrent Information Processes. *arXiv* **2013**, 1–22, arXiv:1210.7784.
- Mikkilineni, R.; Morana, G.; Burgin, M. Oracles in Software Networks: A New Scientific and Technological Approach to Designing Self-Managing Distributed Computing Processes. In Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15), Dubrovnik Cavtat Croatia September 7, 2015; ACM: New York, NY, USA, 2015.
- 9. Burgin, M.; Eberbach, E.; Mikkilineni, R. Cloud Computing and Cloud Automata as A New Paradigm for Computation. *Comput. Rev. J.* **2019**, *4*, 113–134.
- 10. Burgin, M.; Adamatzky, A. Structural Machines as a Mathematical Model of Biological and Chemical Computers. *Theory Appl. Math. Comput. Sci.* **2017**, *7*, 1–30.
- 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.



© 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/).