CHALMERS | UNIVERSITY OF GOTHENBURG

ECSA 2015, Dubrovnik, Croatia
ASDS, Architecting Self-Managing Distributed Systems

A Taxonomy of Computation and Information Architecture

Mark Burgin
Department of Mathematics, UCLA, Los Angeles, USA
mburgin@math.ucla.edu

Gordana Dodig-Crnkovic
Chalmers University of Technology and Gothenburg University,
& Mälardalen University, Sweden
dodig-crnkovic@chalmers.se

COMPUTABILITY AND ARCHITECTURE

A central idea in computability is that of a computational problem which is a task whose computability is explored.

Types of problems:

- Decision problems
- Function problems
- Search problems
- Optimization problems
- Simulation/control of behavior

Basic idea: *computability [our ability to solve a problem computationally] depends on computational architecture.*

PROBLEMS WE WANT TO SOLVE COMPUTATIONALY

- Big data
- Internet of everything
- Semantic web
- Cognitive computing
- Cognitive robotics
- Intelligent cities
- Intelligent homes
- Artificial general intelligence
- Computational neuroscience & neuroinformatics

All above problems depend on the adequate models of computation and they depend on the computational architecture.

WHAT IS COMPUTATION?

The question “What is computation?” is answered differently by different researchers/schools of thought.

Based on the Church-Turing thesis, Turing machine model was long accepted as a universal model of computation, or rather the model of *effective calculability [of a function]*.

WHAT IS COMPUTATION?

The generality of the Turing machine model of computation was questioned on several grounds, by:

- Peter Wegner, through *interaction (adaptivity of computational system) vs. algorithm as a fix procedure*
- Mark Burgin, based on *more general definition of algorithm that does not need to halt*
- Yuri Gurevich*: limitation of TM model as representing a string-to-string computable function*
- Barry Cooper*: TM does not cover higher order computation*
- Samson Abramsky – *computing is about behavior in general*
- …

WHAT IS AN ALGORITHM?

“ (…)

Turing explicated the notion of string-to-string computable function rather than the notion of algorithm.

Even in Turing’s time, the term algorithm had a wider meaning; recall the Gauss elimination procedure or geometric compass-and-ruler constructions. And the meaning of the term algorithm in computer science has been expanding.

(…)

People speak of parallel and distributed algorithms; see [16, 17] for example.

(…)

For us, an algorithm is a (real or imaginable, physical or abstract) computer system at an abstraction level where its behavior—possibly interactive, possibly parallel, etc.—is given or can be given by a program. We devoted much attention to explicating the notion of algorithm [4]. “ Yuri Gurevich

COMPUTATION AS BEHAVIOR

“Traditionally, the dynamics of computing systems, their unfolding behavior in space and time has been a mere means to the end of *computing the function which specifies the algorithmic problem which the system is solving*.

In much of contemporary computing, the situation is reversed*: the purpose of the computing system is to exhibit certain behaviour*. *(…) We need a theory of the dynamics of informatic processes, of interaction, and information flow, as a basis for answering such fundamental questions as: What is computed? What is a process?*

What are the analogues to Turing completeness and universality when we are concerned with processes and their behaviors, rather than the functions which they compute? [16]” Samson Abramsky

COMPUTATIONAL TAXONOMIES

In order to get an idea about how complex the notion of computation is, and that it often means different things to different people, we tried to systematically study computation from different points of view.

EXISTENTIAL TAXONOMY OF COMPUTATION

1. *Physical or embodied (object-based)* computations

2. *Abstract or structural (sign-based)* computations

3. *Cognitive or Mental (interpretant-based)* computations

The above constitutes *layered computational architecture* in cognitive agents. So we could also call it *architectural taxonomy of computation*.

EXISTENTIAL TAXONOMY OF COMPUTATION

1. Physical or embodied (object-based) computations

1.1 Physical computations (as quantum computing)

1.2 Chemical computations (as in chemical morphogenesis)

1.3 Biological computations (information processing in a cell)

2. Abstract or structural (sign-based) computations

2.1 *Subsymbolic* computations - data/signal processing

2.2 *Symbolic* computations - data structures processing

2.3 *Hybrid/mixed* subsymbolic and symbolic computations.

3. Cognitive or Mental (interpretant-based) computation

3.1 *Individual* (computational network of the brain)

3.2 *Group* (computational networks of individuals)

3.3 *Social* (computational networks of groups)

ORGANIZATIONAL TAXONOMY OF COMPUTATION

- *Centralized computations* - where computation is controlled by a single algorithm.

- *Distributed computations* - where there are separate algorithms that control computation in some neighbourhood that is represented by a node in the computational network.

- *Clustered computations* - where there are separate algorithms that control computation in clusters of neighbourhoods.

TEMPORAL TAXONOMY OF COMPUTATION

- *Sequential computations*, which are performed in linear time.
- *Parallel or branching computations*, in which separate steps (operations) are synchronized in time.
- *Concurrent computations*, which do not demand synchronization of separate steps (computations).

REPRESENTATIONAL TAXONOMY OF COMPUTATION

- *Discrete* computations, which include interval computations.
- *Continuous* computations, which include fuzzy continuous processes.
- *Hybrid/mixed* computations, which include discrete and continuous processes.

DOMAIN/DATA TAXONOMY OF COMPUTATION

- The domain of computation is *discrete* and data are *finite*. For instance, data are words in some alphabet.

- The domain of computation is *discrete* but data are *infinite*. For instance, data are ω-words in some alphabet. This includes interval computations because real numbers traditionally are represented as ω-words.

- The domain of computation is *continuous*.

OPERATIONAL TAXONOMY OF COMPUTATION

- Operations in computation are *discrete* and they transform *discrete* data elements. For instance, addition or multiplication of whole numbers.

- Operations in computation are *discrete* but they transform (operate with) *continuous* sets. For instance, addition or multiplication of all real numbers or of real functions.

- Operations in computation are *continuous*. For instance, integration of real functions.

PROCESS-ORIENTED TAXONOMY OF COMPUTATION

- The process of computation is *discrete*, i.e. it consists of separate steps in the *discrete* domain, and it transforms discrete data elements. For instance, computation of a Turing machine or a finite automaton.

- The process of computation is *discrete* but it employs *continuous* operations. An example is given by analogue computations.

- The process of computation is *continuous* but it employs discrete operations. For instance, computation of a limit Turing machine.

LEVEL-BASED TAXONOMY OF COMPUTATION

- At the top and the most abstract/general level, computation is perceived as *any transformation of information* and/or information representation.

- At the middle level, where computation is distinguished as a *discretized process* of transformation of information and/or information representation.

- At the bottom, least general level, computation is defined as a *discretized process of symbolic transformation* of information and/ or symbolic information representation.

SPATIAL (PHYSICAL) LEVELS OF COMPUTATIONS

- *Macro-level* includes computations performed by mechanical calculators as well as electromechanical devices.

- *Micro-level* includes computations performed by integrated circuits.

- *Nano-level* includes computations performed by fundamental parts that are not bigger than a few nano meters.

- *Molecular level* includes computations performed by molecules.

- *Quantum level* includes computations performed by atoms and subatomic particles.

COMPUTATION AS INFORMATION TRANSFORMATION

We are building on our typology of models of computation as information processing (Burgin & Dodig-Crnkovic, 2013).

Future paths for the advancement of the field are expected both as a result of the development of *new computational models (complex computational architectures, concurrent distributed processing)* and learning from nature how to better compute using information transformation mechanisms of intrinsic computation.

As natural cognitive intelligent systems have developed abilities to deal with complexity by efficiently processing data and information, and on a higher level even knowledge (Burgin, 2005) (Burgin, Mikkilineni, Morana, 2015) we are trying to learn from nature how to compute in a more resilient and resource-effective way.

COMPUTATION AS INFORMATION TRANSFORMATION

If we want to further develop computational technologies to solve problems of huge data-processing and information-processing systems on global scale and in real time we should take into account *broader concept of computation than string-to-string mapping*.

One of the approaches going into this direction is *cognitive computing*, that is trying to *mimic human-level cognitive information processing* (“Probably Approximately Correct”- Leslie Valiant).

COMPUTATIONAL ARCHITECTURE OF COGNITION

"(we) propose an *info-computational framework* to approach cognition in living organisms and in embodied cognitive agents of any kind:

the environment affords potential information which the agent can integrate into actual information and transform into knowledge by natural (intrinsic, physical) computation;

perception acts as an information-processing and learning device, through dynamical processes of self-organization of the agent." [Dodig-Crnkovic, 61]

ARCHITECTURE OF COMPUTATION IN LIVING ORGANISMS

In nature the basic info-computational layer is grounded on physico-chemical, chemo-biological and bio-cognitive levels of information processing.

The dynamics of information differs on different levels of granularity of physical processes.

Computation performed by contemporary computing machines (designed computation) is distinctly different from the complex network of networks of computational processes in living organisms (cognitive computation).

CONCLUSION – NEW MODELS OF COMPUTATION

We highlight several topics of importance for the development of new understanding of computation: *natural computation (physical computation), interactivity, concurrency and distributedness* as fundamental for computational modeling of information processing systems such as living organisms and their networks.

The new developments in modeling are needed to support this generalized framework for cognitive architectures. In such a way, we will achieve better understanding of computation as information processing on different levels of organisation in complex cognitive agents.

SOME OPEN PROBLEMS

- How exactly is physical computation (intrinsic/natural computation) connected with abstract computation and cognitive computation?
- How is computation realized in computational systems, in machines, and in living organisms?
- Are computers capable of processing only data or information and knowledge as well, as recent work suggests?
- What can we learn from natural computational processes in cognitive systems that can be useful for engineered information systems and knowledge management, especially when it comes to managing complex systems and big data?