Interactive Paradigms of Computation

Aran Nayebi
Research Seminar in Logic and the Foundations of Mathematics, Stanford University, 4/24/2012
http://www.stanford.edu/~anayebi
“It is absolutely impossible that anybody who understands the question [What is computation?] and knows Turing’s definition should decide for a different concept.”
Strong Church-Turing Thesis

- By the 1960s, it was assumed, as a direct consequence of the Church-Turing thesis, that the Turing machine could completely describe all forms of computation.
- A common misconception (there are many) of the Church-Turing thesis, termed the ‘strong Church-Turing thesis’.
Milner 1993

“Through the seventies, I became convinced that a theory of concurrency and interaction requires a new conceptual framework, not just a refinement of what we find natural for sequential [algorithmic] computing.”
Evolution of Computation

- Many modern applications cannot be adequately described by Turing machines (TMs).
- The interaction between the program and its environment cannot be replaced by any set of inputs determined prior to the computation.
Hypercomputation?

- An emerging movement claiming to present devices that compute non-Turing computable functions or decide Turing undecidable problems.
- These models tend to require “embedded infinities”.
- The most striking claim: Solvability of the halting problem for universal Turing machines.
Hypercomputation? (cont’d)

- Three classes of hypercomputers in a broader sense:
  1. Evolution
  2. Interaction of machines
  3. Infinity
1. Evolution

- Involves adaptation of a TM to either a more knowledgeable component or an infinite sequence of components.
- Upgrade of hardware or software, and may be continuous or discrete steps.
- Can be controlled by interaction with environment, examples including Turing u-machines, site and Internet machines.
2. Interaction of machines

- Involves interaction with an external world, in the form of a singular entity or multiple agents, *during* the computation.
- Examples include Turing’s c-machine and o-machine, and site and Internet machines.
- Sequential vs. distributed interaction
3. Infinity

- Involves not restricting computation to the boundedness of resources
- Examples include allowing an infinite initial configuration, infinitely many computing elements, allowing infinite time, and allowing uncountable alphabets
Feasibility?

- My argument: Hypercomputers that rely on embedded infinities or on a non-recursive oracle to violate, extend, or invalidate the Church-Turing thesis are not physically realizable.
- Interactive models of computation, however, extend the classical sequential paradigm of TMs to adequately describe computing in its contemporary setting.
- These models may provide the first steps of lending physical plausibility to the hypercomputation movement.
Why Interaction?

- One limitation of TMs when it comes to modern computing is accounting for interaction.
- Interaction simply cannot be expressed sequentially, and such models that are more expressive than TMs do not violate the Church-Turing thesis.
- Why? Because the Church-Turing thesis only states that all *effective sequential* computational processes can be simulated by a TM.
Why Interaction? (cont’d)

- Interaction appears in everyday life.
- The notion that interaction requires computational models more expressive than TMs is certainly not recent.
- Under the classical definition of computation, what does the World Wide Web compute?
Models of Interaction

I will now discuss some models of interaction and their practical applications:

1. Site and Internet machines
2. Internet
3. $\pi$-calculus
4. Actor model
1. Site and Internet machines

- Follow the three principles of interaction, evolution, and infinity.
- Site machines: Interactive TMs with advice, where ‘advice’ is a limited version of Turing’s oracle and liberally models machine upgrades.
- Internet machines: A finite but time-varying set of site machines where each machine is specified by its address. Meant to model evolving systems, such as the Internet.
2. Internet

- Internet can operate as a ‘global computer’ which operates on the World Wide Web and interacts with users and geographically distributed resources.
- Grid computing is an example.
- What computational models are appropriate for global computing?
- Answer: For every global TM, there exists a TM with advice that sequentially realizes the same computations as this global TM, and vice versa.
3. \( \pi \)-calculus

- Process algebra built around interaction intended as a successor to the Calculus of Communicating Systems (CCS).
- Based on interaction by message passing and models the changing connectivity of interactive systems.
- More expressive than the \( \lambda \)-calculus.
- However, much closer to a true model of interaction is the Actor model.
4. Actor model

- TMs adhere to boundedness and locality constraints.
- However, the Actor model does not carry out computation in one place; computation distributed in space.
- ‘Actors’ are treated as universal primitives of concurrent computation.
4. Actor model (cont’d)

- Organizational and operational axioms
- Practical applications: Provides a framework for reasoning about many concurrent systems, such as email, various web services (Twitter), and functional programming.
Conclusions

- Strong Church-Turing thesis the result of five incorrect claims:
  1. All computable problems are function based.
  2. All computable problems can be described by an algorithm.
  3. Algorithms are what computers do.
  4. TMs serve as a general model for computers.
  5. TMs can simulate any computer.
Conclusions (cont’d)

These claims can be correctly reformulated as:
1. All algorithmic problems are function-based.
2. All function-based problems can be described by an algorithm.
3. Algorithms are what *early* computers used to do.
4. TMs serve as a general model for *early* computers.
5. TMs can simulate any algorithmic computing device.
6. TMs cannot compute all problems, nor can they do everything real computers can do.
Conclusions (final)

“[In the 70’s] it was premature to openly challenge TMs as a complete model of computation....We believe it is no longer premature to encompass interaction as part of computation” (Goldin and Wegner).

What would be the benefit of such models/theories and how could they be used to advance computer science and modern computing systems?
Selected References


- Hewitt, Carl. “What is computation? Actor model versus Turing’s model”.


The End