COMPUTATION AS SOCIAL AGENCY: WHAT AND HOW

Zhejiang University, Hangzhou, 25 April 2016

Johan van Benthem, Amsterdam, Stanford & Tsinghua, http://staff.fnwi.uva.nl/j.vanbenthem

Abstract The machine paradigm of the 1930s abstracted away from computing by human agents, in a fruitful manner that focused on what is computable in principle. But the agents have struck back. Modern computation is agency in social networks with many communication links, and a wide range of available actions. We discuss the resulting trends under the headings of *epistemization* and *gamification*, linking with philosophy, argumentation theory, social sciences, and game theory. The key logical issue then shifts from the *what* to the *how* of behavior.

Turing Year, Manchester Logic Colloquium, Summer 2012

Turing's achievements, and what remains of them in the Arrow Airport Taxi Company.

Classical themes from the 'golden age'

Universal Machine can do any computation [basis for both practice and basic theory], *Church Thesis:* computation captured completely, *Turing Test*: computation also covers human cognition.

Ongoing debates in US and Europe

Is the Turing model outdated, can we compute a larger class of functions/problems after all?

Logic and fine-structuring views of computation

The rise of programming languages. From computable functions to other levels of structure, say, 'algorithms'? Output and internal structure: *what* one computes vs. *how* one computes: *behavior*.

Modal logic: WHILE programs (Hoare), Dijkstra's structured programming, C+, dynamic logic. Lambda calculus: LISP, functional programming. Divide keeps reemerging in modern versions.

Major challenge around 1980: distributed computation and process theory

Process Algebra (Milner, Bergstra & Klop): bisimulation, trace equivalence... no 'CT' (van Emde Boas on machine models). [Sequential views also alive: μ-calculus, co-algebra of infinite streams.]

New computational paradigms still appearing

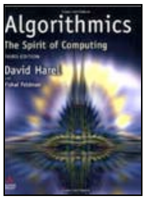
Multi-agent lines: computation as agency, 1980s: Halpern & TARK. 1990s: game semantics, linear logic, Abramsky *Handbook of Philosophy of Information*. Telling detail: communication complexity. Andrew Yao rediscovered von Neumann-Morgenstern Theorem from game theory.

Life after Turing: facebook of logic and computer science

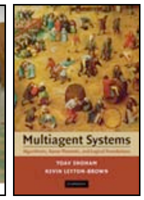
Scott & De Bakker, Kowalski, McCarthy, Dijkstra, Hoare, Pnueli, Milner, Emerson, & others:

Theme today: human metaphor for interactive computation, works both ways

'Computer' from human to machine. Reverse trends ever since: social metaphors of computing, AI & cognition, agent societies of humans and machines, interactive foundations of computation.







Conquering daily life: conversation as computation

Example of the Muddy Children: updates solve the information puzzle, conversation has program structure, sequential and parallel. Dynamic-epistemic logics: again *what* and *how*, model events. Miller & Moss 2005: PAL^* is Π^1_1-complete. [But: complexity of logics ≠ complexity of tasks!]

Daily life strikes back: computation as conversation

Computing is action plus information. Methods can be programs (machine), or plans (human). Two basic features of conversation: *knowledge* [and belief], and multi-agent *interaction*. Suggests exploring transformations from standard algorithms to social procedures. Start with knowledge:

Epistemizing computational tasks

Graph Reachability: Given a graph ***G*** with two points *x*, *y*, does there exist a chain of arrows in ***G*** from *x* to *y*? Solvable in *Ptime* in size of ***G*** (Papadimitriou 1994). The *GR* algorithm performs two tasks: determining if a route exists, and giving an actual *plan* to get from *x* to *y* (see later).

Knowing you made it. You want to reach goal region ϕ, but have only limited observation. $\boldsymbol{G} = (G, R, \sim)$ now has arrows plus epistemic uncertainty links. Brafman, Latombe & Shoham 1993: robot with sensors in-spects current node to see if it is in the goal region: $K\phi$. Kaile Su et al. on sensors inspired logic and epistemology.

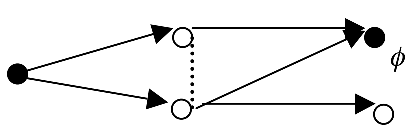
Recent cases of epistemizing: make knowledge explicit in auctions, voting, *social choice*.

Epistemization in general

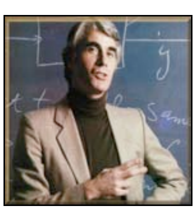
Systematic transformation can be studied as such, few general results known. (Turing himself?)

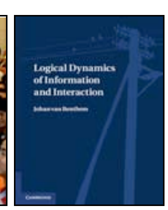
Knowledge programs. Not just epistemize specifications, also algorithms. 'Epistemic programs' (Fagin et al. 1995): test conditions depend on what the agent executing them knows about the facts, or other agents. Connections with imperfect information games. Two roles of knowledge interact:

Having a reliable plan. If we are to trust a plan, should we *know it to be successful*?



After one step, the agent no longer knows if moving *Across* or rather *Up* will reach the ϕ-point. What if we require that $[\boldsymbol{a_1}]K[\boldsymbol{a_2}]K\phi$, where $\boldsymbol{a} = \boldsymbol{a_1};\boldsymbol{a_2}$, etc.? What about more complex programs?



Knowing a program

Know–that versus know–how. New issue: What does it mean to know a program, plan, or method? Attempted reduction to knowledge of effects, dependence on agent types (perfect recall).

Typical informational action: learning. From propositional knowledge goals to 'understanding'. When does an agent understand a program or plan? [What is understanding a mathematical proof?]

Philosophical analyses of knowledge may be relevant. Importance of possibly wrong beliefs and other information-driven attitudes, importance of program revision under new circumstances.

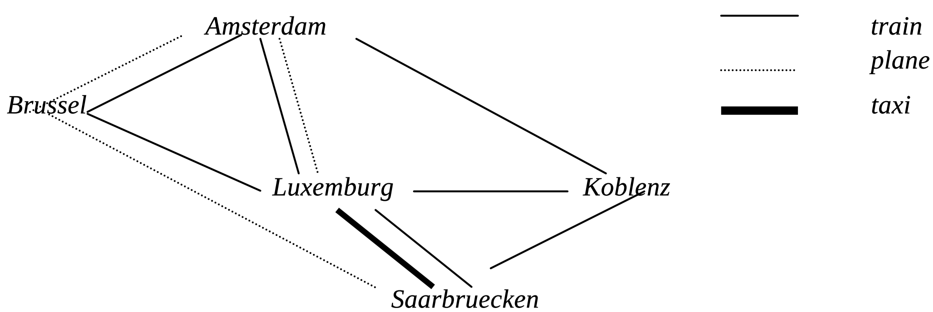
Aside: Various kinds of information are relevant, here not just semantic. Syntax and computing.

Next: Many-mind problems, multi-agent interaction. Gamification *of computational tasks: communication, sabotage. Systematic change in algorithms, from programs to logic of strategies.*

Gamifying algorithms

Reaching a goal comes with variants where *others* should not know. Many subtle algorithms are of social forms [security]. [See also Agotnes & van Ditmarsch 2009.] Much simpler case:

Reachability and sabotage. Consider Graph Reachability in the heart of Western Europe:



What if transportation breaks down, and a malevolent Demon starts disrupting the network? At every stage, let Demon first take out one connection. Now we have a two-player *sabotage game* to be solved. [Current experimental uses as a learning game in cognitive experiments.]

General transformation gamifying any algorithmic task to a *sabotage* game with more players. New logical languages defining these games, and players' strategies. How does *computational complexity* change? For sabotaged reachability, complexity jumps to *Pspace*-complete (Rohde 2005, like Go or Chess). [But cf. Sevenster 2006 on Scotland Yard games, and IF games.]

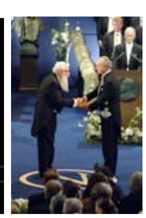
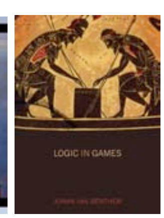
Issue: how does the graph search algorithm itself get 'sabotaged' into a game solution method?

Logic of reasoning about model change can be complex: sabotage modal logic is *undecidable*.

Gamification in general

General program: theory of epistemizing and gamifying algorithms. Links to general *game theory*. Many current contacts between logic, computer science, and game theory. Still larger Facebook:

How vs. *what* again: what are natural levels of fine-structure: When are two games equivalent?

Aside: practical ‘*gamification*’: who invented the term? Theory of this phenomenon wide open.

Recent developments

Logics of social networks. Influences from cognitive science.

Foundational discussions, the three pillars of computation once more

Turing Test: what we need to understand is not ‘computability’, but *behavior*. And the real challenge is understanding behavior in diverse mixed societies of computers and humans.

Church Thesis: from unique ‘what’ to diversity of ‘how’. Process equivalences for agent views. Computation as large family of different styles of behavior. Need base analysis of *social behavior*. A ‘new Turing’: social behavior, suitably conceived, as a universal model for computation?

Universal machine for agency as extended computation: does it exist at all, could it be Games?

References

Johan van Benthem, *2011*, *Logical Dynamics of Information and Interaction*, Cambridge University Press, Cambridge UK. – *2014*, *Logic in Games*, The MIT Press, Cambridge MA. – 2016, ‘Computation as Social Agency’, Strachey Lecture, Department of Computer Science, Oxford, to appear in *Information and Computation*. – 2008, co-edited with Pieter Adriaans, *Handbook of the Philosophy of Information*, Elsevier, Amsterdam.