What Does a Computer Simulation Have to Reproduce? The Case of VMWare

Keith Douglas
Statistics Canada
philosopher.animal@gmail.com

Outline

- Introduction
- Naive answer: functional equivalence
- Semipragmatic answer: Gamer's
- State and event isomorphism
- Obeying same laws
- Answering an objection
- Conclusion

Introduction

- Explanation of virtualization and emulation
- Previous work
- (Additional) philosophical motivation
- Practical motivation

Explanation of Virtualization, etc

- Simulation of computers by computers
- Used to better make use of resources
- Security
- Training

Previous Work

- Douglas 2008, 2010
  - Primarily epistemology

Additional Philosophical Motivation

- Churchland (2007 [2002])

- "More generally, the idea that a machine, any machine, might be programmed to 'simulate' a [von Neumann] machine in particular makes the mistake of treating [von Neumann machine] as if it were itself a special-purpose piece of software, rather than what it is, namely an entirely general-purpose organization of hardware. In sum, the brain is not a machine that is capable of 'downloading software in the first place, and a [von Neumann] machine is not a piece of 'software' fit for downloading in any case."

Practical Motivation

- Self disclosure
  - Not here officially, but work on these problems comes out of my "day job"
- What I do
  - Work as integration specialist / programmer for Canadian federal government
  - VMWare Server and hardened laptop

Naive Answer

- Idea: characterize computer program or environment as a function
- "Logician" answer
- Unsatisfactory:
  - Timing
  - User interface
  - Protocol

Semipragmatic answer: gamer's

- Gist: "To have fun"
- Unsatisfactory:
  - Subjective
  - Can be made intersubjective if every class of applications has experts
  - Area of possible research: take this and generalize to more pragmatic type answers (have been presupposing a weak realism)

Promising answer #1: State and event isomorphism

- Not proposing Popek and Goldberg (1974)
  - P&G do not deal timing and I/O, only deal with same architecture virtualization
  - Microinstructions
- Indifference argument (Makin 1993) suggested by remembering the microinstructions

Promising Answer #2

- Obey same laws: allows overcoming of indifference argument
- Two motivations for this approach:
  - Simulation shares same kind as thing simulated: assume it to judge consequences
  - David Wiggins (2001)

Promising Answer #2 (cont'd): Thesis and virtues

- Thesis: "a simulation of a computer by another computer is successful to the extent that the simulation "obeys" the same laws."
- Virtues:
  - Solve Wordstar problem
  - Solve gamer problem
  - Gives level to which isomorphism (etc.) relevant

Promising Answer #2 (cont'd): Two questions and responses

- Does something have to be of the same ontological category to obey the same laws? (cf. Douglas 2009)
- Doesn't Wiggins' work only apply to natural kinds and their instnaces? (No laws of artefacts qua artefacts.)

Response #1

- In a word, no, the "thingness" of a simulation is guaranteed by the underlying hardware. This is more plausible in the case of virtualization. (Direct instruction execution.)
- Software on guest is software on host
- Subspace relation
- Hence "logician" answer, "isomorphism" not far wrong

Response #2

- Misreading of Wiggins? No, he's just mistaken here: introduce new category "artificial natural" or "constructed natural".
- Not ad hoc: novel chemical compounds, elements. (cf. pentazide ion)
- Laws: computability theory, computational complexity theory (cf. Dean 2007)

Response #2 (cont'd)

- But what ingredients in view of laws?
- Armstrong (1997, 2004), Bunge (1977, etc.)
  - Laws, not law statements
  - Genuine relation between universals
  - "Multiply realizable"
- How much universalism? Open question.
- Stipulate: if narrow view correct, this paper is wrong. (Also, if there are no laws, etc.)

Answer To Reviewer Objection

- I was asked (paraphrased): What if one adopts the view that computers are another "sort of kind"? E.g., hybrid mathematical-natural.
- Fictionalist answer, but:
  - What philosophy of math does this make sense under?
    - Better not turn on something being mathematically describable (cf. structrualism)
    - "Weird" platonism, maybe.
  - "Armstrong-style" physicalist doesn't work. (all is mathematical, not just computers)

Conclusion

- Simulations of computers are like real computers in so far as they "obey" the same laws.
- Follows immediately that, again to they extent that they are "nomologically analogous" they ought to be constrained by the same policies.
- More work needed on idealization and modelling, laws.