

# Defining “Computation”

CPSC471 Yale University  
March 3, 2014

# Objection to Computationalism

## Everything Is A Computer

Something is a computer if someone uses as a computer. End of story. (t.w. Searle, again.)

No, it's not hard to give a satisfactory objective definition. We will discuss David Chalmers's formulation.

## Start with “Computation”

Chalmers asks, “What are the conditions under which a physical system implements a computation?”

Answer: “. . . [W]hen the causal structure of the physical system mirrors the formal structure of the computation.”

Even more precision is coming in a second . . . .

# FSMs

First, define a *finite-state automaton* in a slightly unusual way: There is a set of input states  $I_i$ , internal states  $S_i$  and output states  $O_i$  (all finite sets).

State transitions are of form  $(S, I) \rightarrow (S', O')$ , one for each pair  $(S_i, I_j)$ .

This differs from the usual formulation in that we factor inputs and outputs into separate aspects of the automaton's state.

# Implementation

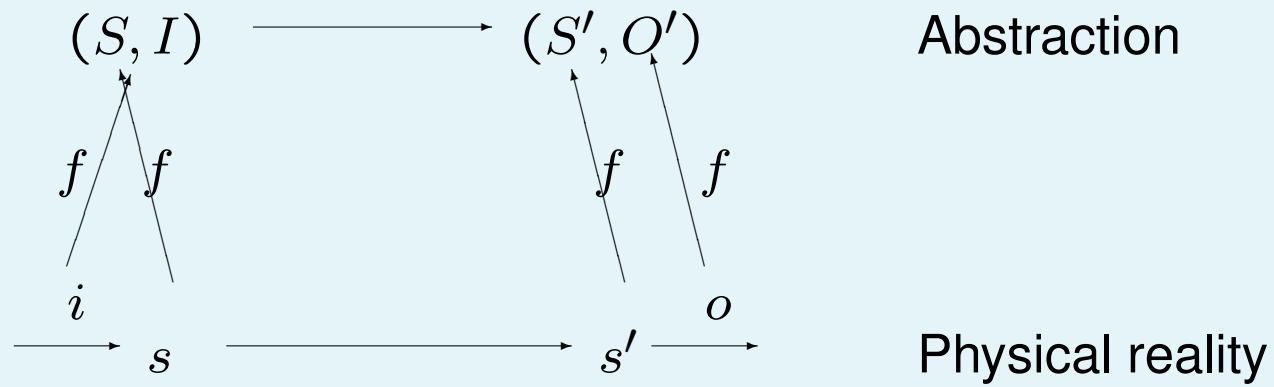
A physical system  $P$  implements an FSA  $M$  iff there is a mapping  $f$

- from internal states of  $P$  to internal states of  $M$ ,
- from inputs of  $P$  to input states of  $M$ , and
- from outputs of  $P$  to output states of  $M$

such that: for every state transition  $(S, I) \rightarrow (S', O')$  of  $M$ , if  $P$  is in state  $s$  and receiving input  $i$  where  $f(s) = S$  and  $f(i) = I$ , this reliably causes it to enter internal state  $s'$  and produce output  $o'$  such that  $f(s') = S'$  and  $f(o') = O'$ .

Note that we allow there to be more (usually many more) than one  $s$  with  $f(s) = S$ .

# Picture Worth 1000 Words



## Types vs. Tokens

Chalmers is not explicit about what “a” computation is? A *kind* of computation (e.g., computing square roots) or a *run* of a system (e.g., the computation of  $\sqrt{5}$  I did yesterday).

He means the former: We’re not identifying the states a system goes through on a particular occasion, but *all* the states it might ever go through.

# Flaws

1. How do you identify the inputs and outputs of a physical system?

Look for eyes and ears?

2. What about *timing*? Could one state transition in the abstract machine correspond to a series of transitions in the physical system?

# Putnam's Theorem

Putnam proves that any physical system implements any FSA.

Sketch of proof: Suppose an FSA goes through state sequence  $ABABA\dots$

Pick a physical system, say, a rock. Suppose over some time period it goes through states  $s_1, s_2, \dots$ . Let state  $A$  be realized as  $s_1 \vee s_3 \vee \dots$ , and  $B$  as  $s_2 \vee s_4 \vee \dots$ . Then the rock implements  $ABABA\dots$

## Chalmers's Response to Putnam

It isn't enough to identify states of a physical system *ex post facto*. It must continue to behave according to the FSA description in the future, especially if its inputs are perturbed in systematic ways.

Putnam's construction collapses under this test regime.

(A system must obey the counterfactual "If  $x$  were in (physical state corresponding to) state  $S$ , it would transition to state  $S'$ .")

# Combinatorial State Automata

Replace the states with vectors.

I don't understand why this is so enormously significant. Since the state spaces of the original formulation could be anything, they could be vectors anyway.

# Issues Chalmers Doesn't Address

At least not in this paper.

1. Noise
2. Approximate computation
3. Continuity

He talks about how you can do Turing machines (sort of), cellular automata, and such, when what we want to know is: Can we model neural networks? Real sensors?

# Objections Refuted

Every system implements many different computations, but not every computation.

# Digestion, Planets, and Computation

Stomachs and solar systems fit his definition of computation (let's suppose). But it's absurd to believe that things like planets decide what to do based on the results of computations.

Yes, it is absurd. But that's not being claimed by anyone. Whereas animals' behavior does depend on the results of computations performed by their brains.

# Relentless Digitalness

Chalmers is relentlessly digital. (As are many other writers.)

I don't think state-transition vocabulary works very well for non-digital systems.

1. Especially if we're trying to shoehorn everything into a *finite-state* formalism.
2. Physical systems are naturally described in the continuous vocabulary of differential equations. For digital systems, this factors nicely into state sets that get lumped together.

# What About Semantics?

Chalmers (quite properly) demurs.

Don't get confused: Neither  $f$  nor its inverse have anything to do with semantics. (Cf. our discussion of Haugeland.)

In my not-best-selling book, *Mind and Mechanism*, I called  $f$  the *decoding*. In your calculator, one decoding maps digits in the display to (surprise) decimal numbers.

The decodings exist even if you don't. There are a zillion decodings, most useless. Whether someone thinks of an decoding or not, a system still implements a computation relative to that decoding. (It's like a frame of reference in physics.)

# So What Is Semantics?

The Census Bureau has a symbol denoting you.

And if they don't, Visa does.

What makes it denote *you*? How does that abstract entity in a computer reach out into the real world and connect to you?

Well, because they have enough information to track you down and in the end surround your house with heavily armed police if you don't pay up.

# Scenario

But suppose some clever weapon wiped out all intelligent life on earth (except you), but Visa's subterranean backup data center kept going.

Would that symbol still denote you?

Many people would say no. Without the users to interpret the symbols, in the end they mean nothing. It might still do computations, described by decodings, but semantics would be gone.

## But Anyway

We're not going to solve, or even discuss, the semantic problem today.

But we don't have to in order to define *computer* and *computation*.

## Discussion Question

What kind of computer is the brain?

Neurons appear to be *asynchronous, continuous* elements. It's possible to build digital computers out of such elements, but not likely that evolution *did*.

But many computations must have digital outcomes. Is there an edge at this point in visual space or not? Should I go left around the boulder or right? Is this sound pattern a voiced or unvoiced consonant?

# Pointers

Does the brain contain anything like pointers?

It's hard to see how it could not. But no one's found a trace of them. (Such a trace would be hard to find.)