Meeting 1: What Is Computation?
Philosophy of Computation at Berkeley
pocab.org
04/14/2017

- Dichotomy of rationalism + romanticism:
    - Is there a special relationship between them?
        - They're the same?
        - It's a rainbow.
        - Algorithmic, elements of randomness that make it deceitful. Essentially they're the same
    - What is beauty?
        - What separates a pile of trash vs a pile of trash in a museum?
            - Looking from it from a different perspective
            - Pile of trash as "pile of trash" -- the idea of a pile of trash -- self-reference makes it artistic
            - Beauty through ideas, juxtaposition
        - The difference is the perspective of finding beauty
            - Art is much more than beauty
            - It's mostly semantics
            - It's philosophy man
            - Separate imagination from beauty, which may mirror the separation of romanticism and rationalism
            - If you're imagining a proof: you have to have some idea of what's real and not real
            - Imagine: run hypothetical situations in your head.
                - Can you say imagination is computation?
                - In that sense yes.
                - Imagination = computation (imagination has infinite steps, and computation is finite steps.)
                    - It's like a STREAM!
                - It's not so much they're the same thing, as it is, computation can emulate imagination and vice versa. (Maybe it occurs after N levels of abstraction.)
                - Does the # of neurons you need in your brain scale with how hard it is to imagine? It's not a question I can answer.
                    - Well after numbers get so big, humans can't really conceive it.

- Well that's more of an unconscious brain. What about the rate of firing a neuron? I can't conceive the rate of firing a neuron.
    - Hofstadter wrote about this: "Are we growing ever number to the ever growing numbers?"
        - Taxpayers don't really know the difference between a million dollars versus a billion dollars.
            - Can you really feel that difference?
            - "I want the difference."
- Last time we talked about abstraction barriers. Denero said if you break it, he'll burn your code.
    - 3 + 3 = 6. This is a very high level of abstraction because you need to look at numbers and then you'll get a result.
    - This is very different from putting together neurons into the brain and firing the neurons and getting into the levels of abstraction to get 6.
    - Most people talk about the human mind is computationally at the level of "3 + 3 = 6" but that can be perceived as very reductionist.
- Secretary Theorems: if you have N things, you can evaluate the value of the things. If you keep it, you can't go to the next one.
    - "You look at the max. But as soon as you keep one, you can't look at the other one?"
    - You're gonna assume a person is this function f(n), the natural value.
    - Didn't Gauss find his wife using this?
    - You interview a certain number of people - you say yes or no. The next one who's better than the first one over e, you choose. If there isn't another one, choose the last one.
    - The n/e is maximized through chances of picking the best one. It's some fraction of n, as it gets larger, tends towards n/e.
- Lovelace says the computer might compose elaborate music/poetry…

- It seems like she's saying something more on a higher level of abstraction.
  - It's sort of like saying, what this machine does, at its core, is composing pieces of music and everything else is based on that.
  - The lower level vs higher level of abstraction is basically, you're building interpreters.
  - If you talk about doing computation on higher level of abstraction, you're saying the brain can emulate a mathematical system.
  - If you're on a lower, you're saying the brain is a mathematical system that can emulate the entity of a mind.
    - I think what she's saying is that, the analytical engine, because it does this computation, can emulate whatever music making machine it can design.
    - You're not saying it's Turing Complete, but it's music composing complete.
- What is knowing what is moral and actually acting moral?
  - If you see something is good, you know it's good.
  - There's some people who will know what's right and wrong and still do wrong.
  - Having empathy versus actually caring about people, is maybe what he means.
  - What's the definition of morality then?
    - It kinda seems to suggest that there's like this 'abstract higher morality.'
      - If I can't reach it because of my limited ideas, I'm not moral.
- Do you need to be good at computer science to be moral?
  - What if you need to write a program to act out a moral situation?
    - Well what if you don't know how to program. That doesn't mean you lack morality.
- A trivial solution: if all the weights are 0. It doesn't give us any more information about what we need to know about the system.
- What's a turing machine?
  - There's a series of chunks. There's an exact reason why this chunk follows from the preceding trunk.

- It implies that computation is a mathematical proof.
- The rules that allow computation don't matter.
- It makes us think if there's any set of rule that is more complex than the bigger set of rules.
  - Church Turing Thesis
- Is a human nothing but a machine that can do computations/a Universal Turing Machine
  - A human is nothing but a Turing Machine, but there's nothing more powerful than Turing Machine so it doesn't matter.
    - "The reason why "nothing but" is a bad thing to say is because before we emulate a Turing Machine, you have to think about it pretty hard. You have to go through the steps in your head that aren't necessarily natural to you. You can do arbitrary computation. But essentially you're implementing a Turing Machine in this much more complicated system in your brain."
      - I'm talking about both higher and lower abstractions.
      - We're sort of drawing this line between regular computation vs. fancy mind things.
      - You're just neurons and your neurons are fancy mind things and your fancy mind things can do computation.
      - Because you can emulate these fancy mind things, with neurons, which are basically just computation (let's assume so). Then it means you can put fancy mind things on top of it again.
        - You can design the computer and work through them yourself, if you have enough space and time.
        - You can build an AI… and it goes on infinitely.
        - A mind is Turing Equivalent because it can simulate anything simulated by a Turing Machine.
      - The definition of computable: computable by a mathematician with infinite pencil, paper, and time.
        - It says that there isn't a physically realizable system that is more powerful than a Turing Machine.
        - We have a physically realizable system using a fancy mind thing using pencils and paper. The pencils and paper provide the fancy mind thing with more space.
        - The pencils and paper take this fancy mind thing and turn it into a fancy mind thing, because you have infinite pencils, paper, and time.

- - - Because you can do a fancy mind thing but these things, you can create as large a fancy mind thing as you'd like.
- We're gonna be starting a blog and we're gonna accept writers!
- What's the difference between humans and animals?
  - Humans are self-referential.
  - Do you think humans are distinct from animals in a definable, qualitative sort of way.
    - A vast majority of animals pass the mirror test.
    - Maybe a lot of animals don't know what they are.
    - The difference between animals and humans is technology.
      - Animals can change their behavior but their behavior modification remains within that generation of change.
      - I object to using culture and technology to differentiate humans and animals because there are animals who have learned how to use spears.
      - What about cultures that don't use the world "me" or "I" in a language (an abstraction and a sense of self)? That doesn't necessarily mean they're not human.
      - There's different ways we can recognize ourselves that are more abstract. Maybe it's just that humans are better than recognizing themselves.
    - Imagine two mirrors: if they're directly facing each other and the image of the mirror are infinitely recursive. Or a TV and a camera.
      - Hofstadter says that humans are fully self-referential but it isn't infinite.
        - Infinite versus approaching infinity?
- The Church Turing Thesis implies that humans are universal Turing Machines
  - Any real world computation can be translated into an equivalent computation involving a Turing Machine.
    - It's more of a matter of FAITH.
    - "I think any amount of reducibility will violate the Church Turing Thesis."
    - Alonzo Church has lambda calculus and he had terms of when something were computable. But when their theories were proven equivalent … it was sorta proved.
      - Well, it kinda hasn't been proved so it does take a bit of faith.
    - There are highly paid mathematicians up in Evans right now doing recursion theory.

- The theorems they prove end up being really beautiful theorems.
- Yeah, but you're trying to start a cult and they don't know anything about math.
    - I would first start with the 'Halting Problem.' Eventually humans can't solve it (idk if that's provable because we don't know how humans work.)
    - We could use an archetype. We could fit the Halting Problem into an archetype.
    - We could say humans can't solve the Halting Problem, but God can.
        - Is God a Computability Oracle?
            -
            - There's an infinite hierarchy of Gods.
- Child Development has something to with computation.
    - The first statement is true because self-reference is necessary for uncomputable.
        - I think so because of Russell's Paradox.
    - Accommodation is necessary for invention
        - Accommodation is analogous to self-reference because you're changing the algorithm.
        - Can it be analogous to learning?
        - The algorithms remain the same, but the weights are different.
            - The neural net as an algorithm changes but the framework in which the works doesn't.
            - We're changing the memories, but the framework.