| | | |
|---|---|---|
| Philosophy 395<br>Computer Science 228<br>Symbolic Systems 210 | **Lecture 3 — Computation in the Wild** | Philosophy of AI<br>Stanford University<br>Winter Quarter, 1989–90 |

| | |
|---|---|
| File path: | Koyaanisqatsi : Philosophy of AI : Phai — 3 — Computation in the Wild |
| Copyright: | © 1990 Brian Cantwell Smith |
| Last edited | Sunday, January 28, 1990 — 7:04:02 PM |

## I. Intro

- A. Where are we?
  - 1. Started, saying "intentionality of computation" is what fuels AI
  - 2. Last time, talked about int'ality, identifying two fundamental properties
    - a. semantic **reach**
      - i. somehow outstrips effective connection to the world
      - ii. because of intentionality's disconnection
      - iii. leads to a tension between the semantical and the effective
    - b. **registration**
      - i. how the intentional phenomenon presents the world as being
      - ii. roughly: meaning
- B. In discussion, promised to talk about logic and model theory
  - 1. Do that today, to start.
  - 2. Get to comp'n in about 20 minutes

## II. Logic

- A. Briefly, start with a language as the int'al phenomenon
- B. Set of sentences
  - 1. N.B.: sentences, not terms, questions, or commands, are paradigmatic semantical units
  - 2. Either atomic or inductively (recursively) defined.
- C. Meant to be a language in which you can say things. Precisely. I.e., an unambiguous "formalism".
  - 1. Cf. George Smith's comment: not meant to be a claim,but rather a uniquely clear language in which things can be claimed.
  - 2. Cf. the calculus in physics.
- D. Syntax.
  - 1. N.B. some problems
      - i. treated abstractly, and
      - ii. not conceptually defined much (though cf. Sylvain Bromberger).
    - a. We'll get back to this when we talk about formality.
  - 2. Proof theory
- E. Model-theoretic semantics

1. Set up a mapping ("interpretation function" or "designation function") between sentences and the model.
2. Whole bunch of issues
    - a. Model isn't necessarily equal to the **subject domain** (world)
        - i. Could be equal; will have more to say
        - ii. Just that it isn't necessarily equal.
        - iii. Quote from Etchemendy:
    - b. Vocabulary
        - i. Model of the sentences vs. model of the world
        - ii. extremely confusing!
        - iii. Will matter, when we get to computation
3. How it's done
    - a. Don't worry about atomic elements: predicates, terms, or even propositions
        - i. Only higher-level relations
    - b. Since you don't know what is true, people who study logic per se, as opposed to those who use logic to write down axiomatisations of things, take on a class of models (with repsect to the full space of possibilities), and then study so-called logical relations on them.
        - i. Such as: tautologies,etc.
        - ii. Examples
    - c. Turns out to be surprisingly illuminating.
4. Then: proofs of convergence
    - a. Soundness, completeness, etc.
5. Some points, for us
    - a. Proof theory: not a full account of inference or anything like that
    - b. Typically contextually independent, etc.

F. So what happened to our two properties?

1. Reach: dealt with implicitly
    - a. Effectiveness: treated abstractly, as syntax
    - b. Disconnection: assumed.
        - i. Utter, according to framework (modulo reflection principles, Gödel structures, etc.)
        - ii. Temporal examples. Not static; rather, no interaction betweeen the "time" of inference and the "time" of the domain.
    - c. Note: semantical relations aren't computed, in any interesting sense of the world.
    - d. But note lesson #1: semantics (content) outstrips syntax (effectiveness).
    - e. Hint: decide the truth of something! What does that mean?
2. Registration. Tricky!

a. To first order, deals with truth and reference (right hand side of last week's list of pairs), not sense or meaning.

b. But: remember it is a model. So these things are often modelled.

c. Three ways:

i. What would make it true in all models <=possible model of meaning

ii. Possible worlds, as an explicit structure

iii. Semantical equations show, rather than state, what the meaning is.

d. Still, some limits

i. No attempt at all to talk about the meaning of the atomic elements.

3. In sum, think it is fair to say that model-theory isn't a very explicit theory of meaning.

G. Morals

1. Limits

a. Modelling

b. No full theory of effective use

c. Lots of particular properties (a-contextual, etc.)

d. Assumes a conceptual scheme.

e. Complete disconnection

f. Entirely abstract

g. …

2. Still, two important lessons (cf. reading)

a. (Repeat): semantics (content) outstrips syntax (effectiveness).

b. Unified theoretical framework, from which both effective and semantical phenomena can be seen and tied together.

## III. Computation in the Wild

A. Intro

1. Start with processes — active, behaving, dynamic, intentional (yes, assume it for now) phenomena.

2. Question is how they're constituted, etc.

3. Notion of a program

a. Three models

i. Specificational (c.s.)

ii. Ingredient (Lisp, AI)

iii. Conversational (Mac, UI, default notion of "language"!)

b. Take specification (as most general) — or, rather, take them all.

c. Talk about **specification**, **ingredients**, ("impressions", in CC), and **interaction**

4. Interpreter, compiler, etc.

a. Interpreter: roughly, a machine whose behaviour is such as to map a program/i into activity.

b. Compiler: maps program/s into program/i.

B. Specification

1. Program/s ⇐ still an intentional phenomenon.

2. Prescription as well as description (Nygärd).

a. cf. recipes: effective

b. (imagine one that wasn't)

c. working drawings for house: build it so that … ⇐ builders weren't used to it.

3. Disconnection issues therefore complicated.

4. Semantics

a. Programming language semantics: study of the relation between a program and the dynamic computation that it engenders.

b. I.e., maps program onto a model of computational processes.

c. ⇒ view that semantics must be effective. Absolutely wrong, in general, as we saw last week! (Promised this.)

d. Model-theoretic!

e. Why model? Well, it's one way to characterise process/insides. So turn to that.

C. Internals

1. Lisp paradigmatically

2. Mental case: this is the subject matter.

a. No reason to suppose there was ever a specification for the human mind.

b. So: realm of thoughts, ideas, thinking, etc.

3. Various ways the "internal structure", as it were, has been handled:

a. Ignored (behaviourism)

b. Functionally (functionalism in $\phi$ of mind)

c. Internal structure

i. By abstract model

ii. Analogy with language

a. "mentalese"

b. Lisp as a language

c. numeral/number confusion (cf. 3-lisp)

iii. Indirect classification

iv. Implementation: same (but assumed) story at lower level. ⇒ operational semantics!

D. Semantics

1. KREP semantics, data bases, etc.

2. semantic value for the semantics of programming languages should be syntactic elements of (true) semantics of computation.
3. This time: surely not effective (depending on your metaphysics)
4. In fact: all the questions of reach, registration, disconnection, etc., apply.
5. Subtlety:
   a. If program/data structure, then can take program to be "about" data structures. But what about them?
      i. Cf. Lisp's original name ("computation of recursive function over uninterpreted symbols").
   b. Other possibilities:
      i. closed world assumption
      ii. use data base as a model or simulacrum (i.e., isomorphic to) the world.
6. ⇒ $1M question
   a. What is the relation between modelling relation (used for analysing the semantics of the program/s) and the true semantic relation? (i.e., Mc, W, and Mw)?
7. Answer
   a. Depends on whether it is one of isomorphism
   b. That's what initial/final algebra approach assumes.

E. So what do we really need?
1. Cross product of four things
   a. specification relation (between program/s and everything else)
   b. internalisation and externalisation (language, etc.) ⇐ haven't talked about this much
   c. implementation (between one level and the next)
   d. semantics (true relation of computational process to the world)
   e. modelling (if that is going to be used to analyse things)
2. ⇒ correspondence continuum

## IV. Conclusion

A. So next three weeks: turn to 3 theories of computation; see how many of these things they deal with, what they have to say, what's right about them (⇐ always crucial), etc.

B. Start with **formal symbol manipulation**. Three handouts:
1. Fodor: "Methodological solipsism" — tough sledding (ϕical)
2. Dretske: "Machines and the mental" — argument that computers can't add.
3. Smith: "Antisemantics" — draft of a chapter of my book

C. Other things:
1. Etchemendy: on model theory
2. Some interpreter terminology

3. Smith: “Correspondence continuum” — §s 4–6 especially on stuff we’ve talked about today.

——end of file ——��