CSci • B607 **Week #1 (a)** **Jan 9, 2001**

 Last edited Thursday, January 4, 2001

Introduction

I. Preliminaries

A. General

1. Welcome!
2. Organisational details (problem sets, grading, readings, etc.) at end of class
3. Handouts
   a. Syllabus (5 pp.)
   b. Reading list (13 pp.)
   c. Lecture notes (this document—8 pp.)
4. Lecture notes
   a. I will prepare full lecture notes, for each class (of which this document is an instance) and post them on the course web site[1] in PDF (both for online reading and for downloading and printing).
   b. Today only, I will distribute paper copies of the lecture notes. In the future I'll assume you can download and print them as needed (before or after class, as you choose).
   c. I will try to post the lecture notes for each week by the *end of the Friday of the preceding week* (in case you want to take an advance look at what we will be covering). If you find me not meeting this deadline, please send me email[2] reminding me—to keep me honest.
5. Readings
   a. There isn't a text, per se, for this course
   b. The primary "text" will be the class notes
   c. In addition, there are a suite of secondary readings
      i. The full set of readings (except for the chapters of my book) will be made available (as soon as possible) as a class reader, for a modest charge
      ii. In the meantime (and later, in addition), copies of the relevant readings (except for papers of mine, including chapters of *AOS*) will be available in a box next to the copier in Lindley Hall • 210,[3] from which you can make your own copies
6. All of my papers (including the AOS book chapters), plus a few of the other papers, will be made available on the class web site, for reading downloading. These include four of the five readings for the first two weeks:
   a. Summary paper: BCS • "Foundations of Computing" (21 pp.)
   b. AOS·I (Introduction) — Chapter 1 • Project (53 pp.)

[1]**http://www.ageofsig.org/courses/b607**

[2]smithbc@indiana.edu

[3]On the second floor, across (and slightly North) from the departmental office.

c. AOS·I (Introduction) — Chapter 2 • Construals (35 pp.)
d. AOS·I (Introduction) — Chapter 3 • The State of the Art (25 pp.)
7. The fifth reading—Haugeland's "Semantic Engines"—is available (for copying) in Sycamore 210 (in a box labeled 'B607') next to the copier.
a. Note: This paper will also be included in the course reader.

B. Background
1. Personal history
a. Undergraduate at Oberlin in the 1960s, torn between sciences and humanities
b. Interested in whether computers could combine:
i. Rigour: of natural sciences
ii. Richness*:* of human condition
c. After 30 years, I think I know the answer. We will get to it by the end of the semester.
2. Will inevitably give the course something of a cognitive science slant.
3. Nevertheless, whereas much of the philosophical and conceptual discussion about the nature of the computing in the 1930s and 40s took place in logic and the philosophy of mathematics, in recent years it has moved almost wholly into the philosophy of mind (Haugeland, Searle, Putnam, Penrose, etc.).
4. While cognitive connections are important, part of our effort is to establish the philosophy of computing as a *legitimate area of specialisation in the philosophy of science*. (Computational issues by now have a far greater impact on surrounding intellectual life than either the (absolute) incompleteness results, or issues in quantum mechanics and relativity.

C. Plan
1. Course will come in 5 parts
2. Part I: Introduction (today + 3 more lectures: 2 weeks total)
3. Parts II–IV: Describe in a moment what else will be covered

II. Project

A. Get at the conceptual terrain by identifying two "promises" made by computing to the surrounding society:
1. **First promise: technological** (pragmatic)
a. Revolutionize life: e-mail, spreadsheets, networked multi-media, etc.
. A phenomenon I call "computation in the wild" (**CITW**)
c. Take it to be the primary topic of mainstream computer science.
2. **Second promise: conceptual** (intellectual)
a. Spread of computation *ideas* into surrounding intellectual life
b. This dispersion has been at least as thorough and rapid as the technology itself.
c. Cf. computational psychology, computational linguistics, etc.
i. UCSD course catalogue: 9 courses called "computational α", for some α

B. Foundational unifications: (structuralism, physics, etc.)
1. Special form of spread of conceptual ideas
2. Given field (¬ computer science) decides (recognizes, intuits, whatever) that to *really* understand the foundations of their field, you need to merge/fuse/unify it with computing.

3. Examples:
   a. Physics: cf. "physics & computing" (or *physics of computing*, or *computational physics*, etc.)
   b. An early one: structuralism in France (cf. Levi-Strauss, Von Neumann, etc.)
   c. Instance of a general rule:
      i. Take what is taken to be central, metaphysical essence of human condition
      ii. Treat that computationally
   d. Two cultural cases:
      i. France: social structures of linguistic and religious practice ⇒ structuralism
      ii. Anglo-America: individual rationality ⇒ cognitive science
4. Summary: computing as an *promiscuous, irresistible bride*

B. Special case: Cognitivism (AI, cognitive psychology, etc.)
   1. "Computational claim on mind" (**CCOM**)
   2. Me: never known what to make of it
   3. Cf. Figure 1.

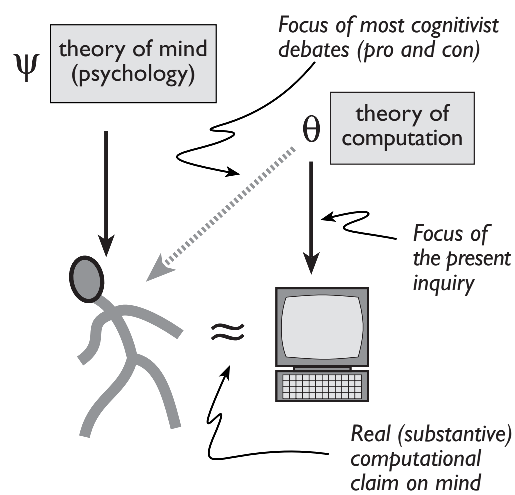

Figure 1 — Computational claim on mind

C. Three criteria
   1. Project: develop a theory of computing that does justice to these developments (1st and 2nd promise, with special reference to cognitivism/CCOM)
   2. Leads to three criteria
      a. **Empirical:** Do justice to computational practice (e.g., be capable of explaining Microsoft Word: the program, its construction, maintenance, and use);
      b. **Conceptual:** Discharge all intellectual debts (e.g., to semantics), so that we can understand what it says, where it comes from, what it costs; and
      c. **Cognitive:** Provide a tenable foundation for the computational theory of mind—the thesis, sometimes known as "cognitivism," that underlies artificial intelligence and cognitive science.
   3. So that's the project:

◆ **(A) Project:** Develop a theory of computation-in-the-wild meeting the 3 criteria

II. Six Construals

A. Don't we already know? Isn't that what computer science, or the "theory of computing," is all about?
   1. Not *no*, but not *yes*, either
   2. Problem: *too many* ideas (not just one)

B. Ideas of two overarching kinds
   1. Primary (we'll go into in detail on these on Thursday)
      a. **Formal symbol manipulation (FSM):** the idea, derivative from a century's work in formal logic and metamathematics, of a machine manipulating symbolic or (at least potentially) meaningful expressions without regard to their interpretation or semantic content;
      b. **Effective computability (EC):** what can be done, and how hard it is to do it, mechanically, as it were, by an abstract analogue of a "mere machine";
      c. **Rule-following (RF)**, or **execution of an algorithm (ALG):** what is involved, and what behavior is thereby produced, in following a set of rules or instructions, such as when making dessert;
      d. **Digital state machines (DSM):** the idea of an automata with a finite disjoint set of internally homogeneous machine states—as parodied in the "clunk, clunk, clunk" gait of a 1950's cartoon robot;
      e. **Information processing (IP):** what is involved in storing, manipulating, displaying, and otherwise trafficking in information, whatever that might be; and
      f. **Physical symbol systems (PSSH):** the idea, made famous by Newell and Simon, that, somehow or other, computers interact with (and perhaps also are made of) symbols in a way that depends on their mutual physical embodiment.
   2. Secondary
      a. *Demeaning*
         i. Computing is *just* … mechanism, machine, artefact, etc.
      b. *Negative*
         i. Computing is … *not* conscious, original, alive, etc.
      c. *Derivative*
         i. Computing is … abstract, universal, formal, …
   3. Others coming along
      a. **Dynamics (DYN)**
      b. **Complex adaptive systems (CAD)**
      c. **Self-organising systems**
      d. Etc.

C. Leads to a more specific interpretation of the project

D. Two phases
   1. **Critical:** for each (primary) construal, determine:
      a. What the construal says or comes to—what claim it makes about what it is to be a computer;
      b. Where it derives from, historically;
      c. Why it has been held;
      d. What's right about it—what insights it gets at;
      e. What is wrong with it, conceptually, empirically, and explanatorily;
      f. Why it must ultimately be replaced; and
      g. What about it should nevertheless be retained in a "successor" more adequate theory.

2. **Constructive**
    a. From these critical studies, assemble what remains
    b. Build a positive account

E. Final part of that project: sketched in a book: *On the Origin of Objects*, MIT Press, 1996.
F. So that's the approach we will take, to the project:

> ◆ **(B) Approach:** Analyse the six major construals, submitting each to seven critical questions, and then—out of the result—constructing a more tenable positive alternative.

III. Substantive Issues

A. Still something missing: we need an *intellectual* guide to this territory. What issues drive it.
  1. Will talk about this Thursday, or next week
  2. Start to get at it in terms of (what we will call) the **mind/body problem for machines**
    a. Physical
      i. Mechanism
      ii. Effectiveness
      iii. ⇒ Body
    b. Semantic, intentional
      i. Meaning
      ii. Reference, representation
      iii. ⇒ Mind
  3. This is a substantive criterion: that any adequate account explain how computation involves, bridges, deals with the physical (effective) and the intentional (semantic).
  4. This problematic will be an organizing theme all semester

B. Dialectics
  1. In fact, the meaning/mechanism (effective/non-effective, "mind"/"body") dialectic is just one of four that play out in the computational realm
  2. All four will be important to us
    a. **Meaning / mechanism** (effective/non-effective, mind/body)
    b. **Abstract / concrete**
      i. Cf. debates about "virtual machines," "levels of abstraction", etc.
      ii. Similarly in cognitive science: functionalism, etc.
    c. **Static / dynamic**
      i. Programs: static, textual entities
      ii. Processes: dynamic, temporal entities
    d. **One / many** (sets, types, properties, groups, etc; but cf. also
      i. Files (multiple copies of one)
      ii. Programs (multiple runs of one)
      iii. Variables (multiple values of one)
      iv. … etc.

C. So that's the substantive focus:

> ◆ **(C) Focus:** In the analysis of computation in the wild in general, and the six major construals in particular, pay particular attention to how each relies on, what each has to say about, etc., the four major dialectics: (i) meaning/mechanism, (ii) abstract/concrete, (iii) static/dynamic, and (iv) one/many.

IV. Formality

A. Finally, there is an issue of *method*
   1. The issue is *formality* (more on this next week)
   2. In brief
      a. Almost everyone thinks that computation is *formal*
      b. But there is enormous difference on what formal means:
         i. Mathematical
         ii. Syntactical
         iii. Rigorous
         iv. Non-semantical
         v. … etc.
   3. Two ways this is held
      a. Substantively: that computation *itself is a formal subject matter, or formal phenomenon*
      b. Methodologically: that computation *should be (or is appropriate for being) studied formally*
   4. One of the main briefs of my investigation has been to see
      a. What might 'formal' mean (in each of the construals)? and
      b. Does that meaning hold of computation-in-the-wild (i.e., is CITW formal, or should it be studied formally, in that sense)?
   5. In brief, the answer will be negative
      a. There is no reading of formality (I will ultimately argue) that is simultaneously
         i. Substantive; and
         ii. True of CITW.
      b. In fact there is an enormous historical irony: computation, thought by many to be the "darling child" of the formal tradition, turns out—*because it is actual*—to contain within it the seeds of that tradition's demise
      c. Like an obstreperous teenager, it cannot be understood by its own forebears.

B. Also: the role of computation in **foundation projects**
   1. Particularly the computational theory of mind (talked about above)

C. So that's the methodological focus:

> ◆ **(D) Method:** In analysing computation in the wild in general, and the six major construals in particular, pay attention to the role of *formality*—whether computation is formal, whether it is (or warrants being) studied formally, etc.

V. Summary

A. So that's the project:
1. (a) Project
    a. Look at each of six construals
    b. With respect to all 3 criteria (empirical, conceptual, cognitive)
2. (b) Strategy
    a. For each, answer the questions above, wrt its treatment of computation in the wild
    b. With special emphasis on
        i. (c) How it deals with four dialectics
            α Meaning/mechanism (effective/non-effective, mind/body)
            β Abstract/concrete
            χ Static/dynamic
            δ One/many
        ii. (d) What it says about formality
3. (e) Construction: when all is said and done: assemble the parts for a new theory

VI. Structures of the course

A. Lectures
1. **Part I • Introduction** — 2 weeks
    a. (Thursday) Go through the six construals in a little more detail. Get a feel for what they say, what kinds of issues they bring up, what is at stake in each one, etc.
    b. (Next Tuesday) Four dialectics, especially the mind/body problem for machines
    c. (Next Thursday) Programs, processes, implementation, and architecture
    d. Then turn to three construals
2. **Part II: Formal Symbol Manipulation (FSM)** — 4 weeks
    a. Note: will be familiar to cog sci, AI, phil mind, etc.
    b. Also to logicians
    c. *Not* to computer scientists
    d. Need to do first, because gets semantics onto the table
3. **Part III: Effective Computability (EC)** — 4 weeks
    a. Inverse: familiar to cs; only slightly less so to cog sci, AI, phil mind
    b. One highlight of the course: solve the halting problem!
    c. Will demonstrate a simple machine that solves the halting problem as it is normally stated.
    d. Obviously, doesn't *really*. In fact it is a cheat.
    e. Our task will be to track down *why it is a cheat* (and draw the appropriate morals).
4. **Part IV Digital State Machines (DSM)** — 2 weeks
    a. Hardest—and most consequential (deepest challenge to formalism, analytic tradition)
    b. Also lots of fun.
5. **Part V • Conclusion**

B. Readings

1. Books: *The Age of Significance*
   a. Draw picture of seven volumes.
   b. Will hand out chapters of first 3 volumes (picture)
   c. Note: current versions are *very old, and not very crisp*. The lecture notes will be better.
   d. Also: paper ("Foundations of Computing"): condensed summary of the whole course.
2. Readings
   a. For intro: use my books and papers. Also Haugeland
   b. For FSM and EC: volumes 2 and 3, with backup papers
   c. For DSM: primary papers
3. Distribution
   a. AOS: PDF on the web site (download and print).
   b. Others: Copies will be placed in a box marked "B607" in the computer science department copy room (Lindley 210)—for you to make copies.

C. Assignments

1. Series of (14) Problem Sets
   a. One per week
      i. In general, they will be due on Tuesday, by class time.
      ii. The first (small) one will be handed out on Thursday, due next Tuesday (January 16)
   b. Each one will be a structured essay questions (guided step-by-step paper)
   c. Will take at least 4–5 hours to work through.
2. Starting next week, each problem set will be posted on the class web site by Friday, 10 days before it is due.
3. Posting
   a. Each person—or each group, or sub-group[4]—should post their response on the class web site, using the **Annotate** software,[5] written by Ruth Eberle (of IU cognitive science).
   b. I will read and comment on each person's answer on line (using paragraph-by-paragraph annotations). You are free (encouraged) to read other people's answers, make your own comments, etc.
4. Collaboration
   a. I *recommend* that you work with others (really want to open up a dialogue).
   b. Responses may be submitted individually (in which case they should be written individually), or else by groups—or subgroups—in which case they should list the collaborators, and then, if appropriate, indicate who is responsible for what points or insights or work, if substantially differential credit is due, at different parts of the reply.
   c. The one requirement: own up (when submitting work) to whom you have depended on, and for what (just as you would in a professional paper).

—— *end of file* ——

[4]See the remarks about collaboration in the syllabus.

[5]**http://www.indiana.edu/~annotate**