B607 Readings (Provisional) Spring 2001

 Last edited Monday, January 8, 2001

Notes

1. All the readings listed here, including those labeled "primary," are in a certain sense supplemental; the main reference work, for the course, will be the lecture notes—available for browsing, downloading, and printing on the class web site (**http://www.ageofsig.org/courses/b607**).
2. This reading list is something of a work in progress; entries may be updated as the course proceeds. Many important papers appeared in the last few years (e.g., on Putnam's "open system" theorem), and several journals (*Synthèse*, *Minds & Machines*) have devoted special issues to the nature of computing. Some selections from these have been included below, but a reworking of the entire bibliography is due. Finally, we may add papers of special interest to class members.
3. A (partially annotated) web-based bibliography on the philosophy of computing is under development. As soon as possible a pointer to it, with a revised list of suggested readings and links, will be posted on the class web page.
4. Part I (Summary) lists only authors and titles. Full references, together with short summaries / commentaries, are given in Part II.

I • Summary

I. Introduction (2 weeks)

A. Summary
   1. Smith, Brian Cantwell, "The Foundations of Computing"

B. Primary
   1. AOS·I (Introduction) — Chapter 1: "Project"
   2. AOS·I (Introduction) — Chapter 2: "Construals"
   3. AOS·I (Introduction) — Chapter 3: "State of the Art"

C. Secondary
   1. Haugeland, John, "Semantic Engines"

II. Formal Symbol Manipulation (4 weeks)

A. Primary
   1. AOS·II (Formal Symbol Manipulation) Chapters 1–4

B. Secondary
   1. Background
      a. Hunter, Geoffrey, Part I §s 1–7 of *Metalogic: An Introduction to the Metatheory of Standard First Order Logic*.
   2. Computation as formal symbol manipulation

a. Hayes, Patrick J., "Computation and Deduction"
b. Kowalski, Robert, "Algorithm = Logic + Control"
c. Newell, Alan and Simon, Herbert A., "Computer Science as Empirical Inquiry"
d. Newell, Alan, "Physical Symbol Systems"

3. Analysis, discussion, and critique
   a. Fodor, Jerry A., "Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology"
   b. Fodor, Jerry A, and Pylyshyn, Zenon, "Connectionism and Cognitive Architecture: A Critical Analysis"
   c. Dretske, Fred I., "Machines and the Mental"
   d. Searle, John "Minds, Brains, and Programs"
   e. Haugeland, John, "Syntax, Semantics, Physics"

III. Effective Computability and Recursion Theory (5 weeks)

A. Primary
   1. AOS·III (Effective Computability) Chapters 1–4
   2. AOS·I (Introduction) — Chapter 6: "Synopsis • I"

B. Secondary
   1. For Turing machines themselves
      a. Minsky, Marvin, Chapters 5–8 of *Finite & Infinite Machines*
      b. Turing, Alan M., "On Computable Numbers, with an application to the Entscheidungsproblem"
      c. Turing, Alan M., "Computing machinery and intelligence"
      d. Kleene, Stephen C., "Turing's Analysis of Computability, and Major Applications of It"
   2. For discussion (historical)
      a. Gandy, Robin, "The Confluence of Ideas in 1936"
      b. Davis, Martin, "Mathematical Logic & the Origin of the Modern Computer"
      c. Webb, Judson, Introduction & Chapter 1 of *Mechanism, Mentalism, and Metamathematics: An Essay on Finitism*
      d. Gandy, Robin, "Church's Thesis and Principles for Mechanisms"
   3. Analysis, discussion, and critique (recent)
      a. Searle, John, chapter 9 from *The Rediscovery of the Mind*; or "Is the Brain a Digital Computer?"
      b. Putnam, Hilary, appendix to *Representation and Reality*
      c. Chalmers, Dave, "Does a Rock Implement Every Finite-State Automaton?"
      d. Copeland, B. J. (1996) "What is Computation?"
      e. Scheutz, Matthias, chapters 2–4 from his dissertation "The Missing Link: Implementation and Realization of Computations in Computer and Cognitive Science"

IV. Digital State Machines (3 weeks)

A. Primary
   1. AOS·I (Introduction) — Chapter 7: "Synopsis • II"

B. Secondary
   1. For the notion of a digital state machine
      a. Minsky, Marvin, Chapters 1 & 2 of *Finite & Infinite Machines*
   2. For the notion of digitality
      a. Haugeland, John, Chapter 2 of *Artificial Intelligence: The Very Idea*
      b. Goodman, Nelson, Chapter 4 of *Languages of Art*
      c. Lewis, David, "Analog and Digital"
      d. Haugeland, John, "Analog and Analog"
      e. Dretske, Fred, "Sensation & Perception"; chap. 6 of *Knowledge & the Flow of Information*
      f. Fodor, Jerry A. & Ned J. Block, "Cognitivism & the Analog/Digital Distinction"

V. Information Processing (to be skipped)

A. Primary
   1. AOS·I (Introduction) — Chapter 7: "Synopsis • II"

B. Secondary
   1. For the syntactic notion
      a. Weaver, Warren, "Recent Contributions to the Mathematical Theory of Communication"
      b. Shannon, Claude E., Part I of "The Mathematical Theory of Communication"
      c. Singh, Jagjit, Chapters 1–9 of Great Ideas in Information Theory, Language, and Cybernetics
   2. For the semantic notion
      a. Dretske, Fred I., "Précis of *Knowledge and the Flow of Information*"
      b. Dretske, Fred I. Chapter 3 of *Knowledge and the Flow of Information*
      c. Israel, David and John Perry, "What is Information?"
   3. For application of the semantic notion to AI and computer science (respectively)
      a. Rosenschein, Stanley J., "Formal theories of Knowledge in AI and Robotics"
      b. Halpern, Joseph, "Using Reasoning about Knowledge to Analyze Distributed Systems"

VI. Conclusion (The Age of Significance) (1 week)

A. Primary
   1. [AOS·I (Introduction) — Chapter 6: "Synopsis • I"]
   2. [AOS·I (Introduction) — Chapter 7: "Synopsis • II"]
   3. AOS·I (Introduction) — Chapter 8: "The Middle Distance"

II • Full References and Annotations

Note: Papers and chapters by the instructor (BCS) are not listed here; drafts will be made available (for downloading and printing) from the class web site.

I. Introduction

1. Haugeland, John, "Semantic Engines", in Haugeland, John (ed.), *Mind Design: Philosophy, Psychology, Artificial Intelligence*, Cambridge, Mass: The MIT Press/a Bradford Book (1981), pp. 1–34.
   - The introductory essay to an excellent collection of papers on the philosophy of AI as of ~1980 (before the rise of connectionism, situatedness, non-representational robotics as inspired by phenomenology, ethnomethodology, etc.). Although the paper is framed in terms of AI and philosophy of mind, it is a clear and plain introduction to the notion of a computer as an automatic formal digital system with an interpretation. Three of those words—*formal*, *digital*, and *interpretation*—will be big ticket items later in the course. Haugeland's notion of formality is idiosyncratic, though, as we will discuss in the critical part of Part II. Summary: essential background reading for anyone interested in the philosophy of computation and/or AI.
   - Note: This introduction is to the first edition of *Mind Design*, rather than the current version, *Mind Design II* (also MIT Press). The latter version's introduction covers much of the same material, but in a more compact—and, in my view, less accessible—fashion.

II. Formal Symbol Manipulation

A. Background

1. Hunter, Geoffrey, §s 1–7 of "Part One: Introduction: General Notions" of *Metalogic: An Introduction to the Metatheory of Standard First Order Logic*, Berkeley and Los Angeles, CA: University of California Press (1971), pp. 1–15.
   - An extremely simple and abbreviated summary of some of the most standard notions of formal logic ("formula," "symbol," "deduction," "model theory," etc.). Included as essentially a cheat sheet. For more detail one can read more of Hunter, or any of a number of other introductory texts (Hodges, Crossley, Jeffrey, Mates, Mendelson, Salman, Barwise & Etchemendy, etc.—ask me if you are interested).

B. Computation as formal symbol manipulation

1. Hayes, Patrick J., "Computation and Deduction", *Proceedings of the 2nd MFCS Symposium, Czechoslovak Academy of Sciences*, 1973, pp. 105–118.
   - A somewhat obscure paper that stands midway between the notions of computation as deduction derived from the recursion-theoretic tradition (which we will get to in section V) and the explicit programming community in computer science. It was way ahead of its time in foreshadowing the development of Prolog. Note that Hayes is much clearer on the difference between proof and interpretation than many who followed him in the "logic programming" camp. Historically important.
2. Kowalski, Robert, "Algorithm = Logic + Control", *Communications of the ACM*, 22 (July 1979), pp. 424–436.

- A more famous paper than Hayes's, but obviously in part inspired by it (Kowalski not only refers to Hayes, but in his paper Hayes refers to two joint papers with Kowalski, dated 1969 and 1971). Both authors are concerned to get the issue of control into the ring to be treated as a subject matter in its own right, in a way that logic itself does not do, and that Prolog, often thought to be the natural heir of this work, also does not quite do. Also, in spite of Kowalski's claim (in his first ¶) that he is not concerned with the use of predicate logic as a programming language in its own right, he nonetheless conceives of programming in a very logic-like way. Something to think about: is it *programming* that Kowalski is analogising to logic (+ control), or computation itself?

3. Newell, Alan and Simon, Herbert A., "Computer Science as Empirical Inquiry", *Communications of the Association for Computing Machinery*, 19 (March 1976), pp. 113–126. Reprinted in Haugeland's *Mind Design* (1981), pp. 35–66.
   - Newell and Simon's Turing Award paper, in which they summarise their approach to computation in the "physical symbol system hypothesis" (PSSH). Although discussed here in AI and cognitive psychology terms, it is included here because the PSSH is so often associated with the formal symbol manipulation view. For a more detail version, see #6, below.
4. Newell, Alan, "Physical Symbol Systems", *Cognitive Science* 4:2 (1980), pp. 135–183.
   - Another famous paper, which presents a more detailed story about Newell and Simon's notion of what it is to be a symbol system. Although the machine-level details can be ignored, it is instructive to observe how Newell understand symbols, designation, interpretation, and other semantical notions.

C. Analysis, discussion, and critique

1. Fodor, Jerry A., "Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology," *The Behavioral and Brain Sciences*, **3**:1 (March 1980), pp. 63–109. Reprinted in Haugeland (1981), pp. 307–338, and in Fodor, Jerry A., *RePresentations: Philosophical Essays on the Foundations of Cognitive Science*, Cambridge, Mass: The MIT Press/a Bradford Book (1981), pp. 225–253.
   - A classic paper in the philosophy of psychology, articulating the "formality condition" on mind, which Fodor takes to be constitutive of what it is to be computational. Non-philosophers may find this tough going, but if read carefully (don't be distracted by the cuteness) it is pretty sensible. Note that Fodor is absolutely committed to the existence of mind–world semantic relations; he just doesn't think they can be captured in a computer, and hence that they are not an appropriate subject matter for AI (and cognitive psychology). When Fodor presented this paper at the AI Lab at MIT in the 1970s, many in the audience (after wrestling with his philosophical style) said "hey, yeah, that's exactly right; of course there's nothing but syntax and formality." Fodor was appalled.
2. Fodor, Jerry A, and Pylyshyn, Zenon, "Connectionism and Cognitive Architecture: A Critical Analysis" *Cognition*, **28:** 3–71 (1988); reprinted in Steven Pinker and Jacques Mehler (eds.), *Connections and Symbols*, Cambridge, Mass: MIT Press/a Bradford book (1988); and as chapter 12 of John Haugeland (ed.), *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Cambridge: MIT Press 1997.
   - Fodor's ultimate defense of the traditional "language of thought" (LOT) theory of mind against the onslaught of the connectionists. This is an important paper, above and beyond its importance in the LOT/connectionist debate, because in it Fodor pares down,

to bare bones, his notion of a "language" of (mental) representation. As such, the paper provides a distilled version of the traditional computational theory of mind.

3. Dretske, Fred I., "Machines and the Mental", Presidential Address to the 83rd Annual Meeting of the Western Division of the American Philosophical Association, *APA Proceedings*, Chicago (1985), pp. 23–33.
   - A very nice short paper in which Dretske's argues that computers can't add. While there are many reasons to disagree with him (ultimately I believe he is wrong), he is closer to the mark than many computer scientists will think. The best way to read this paper is not only to see whether or not you agree with him (and if so, why), but to ask whether or not his conclusion is implied by the formal symbol manipulation view that he embraces (and that is advocated by Hayes, Kowalski, and other logicists, as well as many other philosophers of mind).
4. Searle, John, "Minds, Brains, and Programs," first published in *Behavioral and Brain Sciences*, **1:** 417–424 (1980), © Cambridge University Press; reprinted as chapter 7 of John Haugeland (ed.), *Mind Design II: Philosophy, Psychology, Artificial Intelligence*, Cambridge: MIT Press 1997.
   - The classic paper on the Chinese Room—a thought experiment in which Searle tries to use the formality condition (the thesis that computational operations are independent—or anyway proceed independently—of semantics) to argue that people, or brains, cannot be computational. In spite of its considerable fame, Searle's argument has, to my knowledge, convinced no one. Yet it has proved surprisingly difficult, many critics would agree, to say just what it is that Searle fails to understand.

     My own diagnosis is that the situation betrays the inadequacy of extant conceptualisation. Searle would be right, I maintain, if the "formal" construal of computation he assumes were true. The fault, therefore, (in at least a certain sense) remains in computer science's court. What (we) computationalists need to do is to articulate, in terms that a non-computationalist can understand, what it is to be computational, in such a way that Searle's deviant conclusions can be blocked.
5. Haugeland, John, "Syntax, Semantics, Physics" (ms., forthcoming)
   - One of many replies to Searle,[1] that deals both with Searle's first argument against the computational theory of mind (the Chinese room argument, above, that semantics is independent of syntax), and with his second argument (III.C.1, below), that syntax is independent of physics. One thing that makes Haugeland's analysis especially powerful is that it adopts the (stronger, in my view) strategy of explicitly denying Searle's basic understanding of computing, rather than arguing that his conclusions don't follow from those assumptions (a weaker, and, in my judgment, ultimately untenable, strategy).

III. Effective Computability and Recursion Theory

A. For Turing machines themselves
   1. Minsky, Marvin, Chapters 5–8 of *Finite & Infinite Machines*, Englewood Cliffs, N.J.: Prentice-Hall (1967), pp. 103–156.
      - As in II.A.1, above, these chapters are included as presumed background. They provide a clear, simple, and relatively non-mathematical introduction to the notion of a Turing machine, including the universal Turing machine, the idea of a decision procedure, etc.

[1] For replies to Searle's first argument, see the commentaries in the *Behavioural and Brain Sciences* source of II.C.4.

2. Turing, Alan M., "On Computable Numbers, with an application to the Entscheidungs-problem", *Proceedings of the London Mathematical Society* (series 2) **42** (1936–37), pp. 230–265. A correction, ibid. **43** (1937), pp. 544–546. A useful critique appeared as an appendix to Emil L. Post's "Recursive unsolvability of a problem of Thue", *Journal of Symbolic Logic*, **12** (1947), pp. 1–11. Both the paper and Post's appendix are reprinted in Martin Davis, (ed.), *The Undecidable. Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Hewlett, NY: Raven Press (1965), pp. 116–154 and pp. 299–303, respectively.
    - The classic paper, which any serious student must know. If one ignores the details of the coding (which neither matter to the conceptual structure of what is going on, and of course could now be done much more simply and perspicuously) the argument is not as intimidating as it may seem at first. Two notes: (i) by "computer" Turing meant a person; he uses "machine" or "computing machine" for what we now call the computer; (ii) by "decimal" in the first sentence he essentially means a positional-notation numeral with a "decimal point" (his coding schemes are often binary).
3. Turing, Alan M., "Computing machinery and intelligence," *Mind* **LIX**: 236 (1950) pp. 433–460; reprinted in Alan Ross Anderson (ed), *Minds and Machines*, Englewood Cliffs, N.J.: Prentice-Hall (1964), pp. 4–30.
    - The other Turing classic. This is the place where the Turing test is introduced and discussed. Although the test itself, and the surrounding discussion of intelligence, belongs in a companion course on the philosophy of artificial intelligence or cognitive science, this paper, being more conceptual and speculative than #26, better conveys Turing's own understanding of the digital computing machines he had invented.
4. Kleene, Stephen C., "Turing's Analysis of Computability, and Major Applications of It", in Rolf Herken (ed.), *The Universal Turing Machine: A Half-Century Survey*, Oxford: Oxford University Press (1988), pp. 17–54.
    - Kleene's summary and analysis of Turing's work, a little more mathematical than #25 (though still pretty elementary as such things go!). Interesting for its clarity, for its authority, and for some of the historical remarks scattered through it, which convey a sense of the people and the times as well as the ideas. As its name implies, the Herken volume, from which this and the next two papers are taken (out of a total of 28), both celebrates and reflects on the history of computability theory.

B. For discussion (historical)

1. Gandy, Robin, "The Confluence of Ideas in 1936", in Herken (1988) (see #28), pp. 55–111.
    - Although Gandy is no novelist, this is nonetheless a fascinating picture of what was happening in the 1930s, who the major players were, how the Church-Turing thesis arose, etc. (it even starts with a section on Babbage). Necessary reading for anyone interesting in placing the technical developments in their appropriate historical and conceptual context.
2. Davis, Martin, "Mathematical Logic & the Origin of the Modern Computer", *Studies in the Working of Mathematics*, The Mathematical Association of America ( ), reprinted in Herken (1988) (see #26), pp. 149–174.
    - Davis is not only himself a recursion theorist, but has been an expositor of this tradition of work, having written a variety of books and edited several collections on the subject.

Like #29 this takes an historical attitude, but is less concerned with the microevents of 1936 than with how those years contributed to computing more generally. Very much a mathematician's approach, not a computer scientist's (the paper also ends rather unsatisfyingly, with a rather wimpy final paragraph on developments since 1945). Still worth reading.

3. Webb, Judson, Preface, Introduction, and Chapter 1 ("Mechanism: Some Historical Notes") of *Mechanism, Mentalism, and Metamathematics: An Essay on Finitism*, (Synthèse library; v. 137) Dordrecht: D. Reidel (1980), pp. vi–xii and 1–32.
   - This is a difficult chapter; probably the most difficult reading in this whole list, in spite of the fact that it is the opening chapter of an entire book. Fascinating reading, though, for those with the patience to work it through — not only for its historical interpretation of computability, the undecidability results, etc., but also for its striking claim that the incompleteness and undecidability results ultimately argue *for*, not against, the thesis of mechanism. This book, and also Earman's *A Primer on Determinism* (…), point the way towards much thorough analyses of the philosophical consequences of computability than we can get to in this course.
4. Gandy, Robin, "Church's Thesis and Principles for Mechanisms," in Jon Barwise, H. Jerome Keisler, and Kenneth Kunen (eds.), *The Kleene Symposium*, North-Holland Publishing Company (1980) 123–148.
   - A demanding paper that attempts to derive the formal (syntactic) constraints on Turing machines (constraints that Turing essentially stipulated, as presumptively reasonable) from four elementary, but conceptually rigorous, notions of what it is to be a mechanism. Especially impressive is Gandy's demonstration that even rather subtle adjustments to the four principles have ramifications that affect the notion of computation resting on them (more specifically: that the four principles must be formulated in a very precise and particular way in order for the Turing results to follow).

     Open questions: (i) Does Gandy's analysis ground the notion of computing in ways that Searle and Putnam (III.C.1 and III.C.2, below) would accept? (ii) Can Gandy's principles be derived from (classical) physics? (iii) To what extent does Gandy's construction demonstrate—to what extent can it be relied upon to show—that, because it does not satisfy these four basic constraints, quantum (and perhaps other kinds, such as DNA or organic?) computing could yield a notion of "what is computable" outside the normal computability limits?

C. Analysis, discussion, and critique (recent)
   1. Searle, John, chapter 9 from *The Rediscovery of the Mind*, Cambridge, Mass: The MIT Press (1992); an earlier version was presented under the title "Is the Brain a Digital Computer?" as an APA Presidential Address in 1990.
      - Searle's second argument against the (strong) computational theory of mind, resting on a thesis that syntax is independent of physics—from which, according to Searle, it follows that syntax is an observer-relative, rather than intrinsic, property, and thus irrelevant to scientific explanation. Searle thinks that this argument is just as decisive as his first argument (the famous "Chinese room" argument that semantics is independent of syntax—see II.C.4, above).
   2. Putnam, Hilary, appendix to *Representation and Reality*, Cambridge, Mass: The MIT Press (1988).

- Putnam's argument that every "open system" implements every finite-state automaton. Putnam uses this theorem to argue against functionalism (the theory of mind that he originally formulated,[2] likening the mind to a Turing machine), but for our purposes it is of direct interest to the notion of computation, independent of whether than notion is applicable to mind. Specifically, it highlights the difficulties that follow from being able to understand physical systems in terms of arbitrary, wildly complex, physical state types (or, equivalently, from being able to "assign" such types to concrete mechanisms).

  Putnam's theorem generated a flood of replies, three of which are listed below. See in addition Chalmers, D. J. (1994) "On Implementing a Computation", *Minds and Machines* **4:** 391–402, and Chrisley, R. L. (1994) "Why Everything Doesn't Realize Every Computations", *Minds and Machines* **4:** 391–402.

3. Chalmers, Dave, "Does a Rock Implement Every Finite-State Automaton?", *Synthèse* **108:** 310–333.
   - An argument that Putnam's theorem (above) fails because, according to Chalmers, the types that Putnam assigns aren't modally strong enough. Chalmers believes that the state type assignments in terms of which computational explanations are given should be strong enough to support claims about what a system *would have* done, had circumstances been different. While this desideratum seems reasonable, it is not clear that Putnam's constructions are actually vulnerable to it; see Scheutz's analysis, below. In my own judgment, Scheutz is right on two counts: (i) that Chalmers argument against Putnam fails, and (ii) that Chalmer's positive proposal ultimately fares no better than Putnam's own.

     So far as I know, the basic problem—of how to restrict possible physical state types—remains unsolved (though see Scheutz's pragmatic proposal, below; and Gandy's analysis in III.B.4, above).
4. Copeland, B. J. (1996) "What is Computation?" *Synthèse* **108:** 403–420.
   - Another reply to Putnam, countering his construction with a proposal to understand computing in terms of a set of labeling functions. In my view, two concerns limit support for this paper's conclusions: (i) the overly epistemological character of the proposal—a worry that it confuses ontological facts about what it is to be a computer (or be computational) with epistemic facts about how we characterise such systems; and (ii) a worry about the exclusively syntactic (non-semantic) character of computing it endorses.
5. Scheutz, Matthias, chapters 2–4 from his dissertation *The Missing Link: Implementation and Realization of Computations in Computer and Cognitive Science*, Indiana University Dept. of Computer Science, Bloomington, Indiana, September 1999.
   - An analysis and tying-together of Searle's (III.C.1) and Putnam's (III.C.2) arguments that computational typings of systems are too unconstrained to serve as the basis for scientific explanation, and a critique of both Chalmer's and Copeland's attempted replies. Scheutz argues, using what he calls a "slicing a theorem", that a strengthened version of Putnam's construction survives all such challenges, and establishes that unless the notion of a physical type can be restricted, no higher-level constraint (e.g., on the notion of syntax) can be formulated so as to yield a substantive notion of computation. *(cont'd)*

---

[2]See Putnam, Hilary, "The Meaning of 'Meaning'", in *Mind, Language, and Reality—Philosophical Papers, Vol. 2.* Cambridge, England: Cambridge University Press (1975).

Scheutz's own proposal is to abandon the search for a foundational reconstruction of a substantial notion of physical type, and to start instead with the sorts of practical typings that engineers use in their routine practice. Pacé considerable sympathies with pragmatist metaphysics, I think we can still hope for something better: an intellectually satisfying and ontologically robust and substantial notion of a physical type. But I agree with Scheutz that such an account remains to be given.

IV. Digital State Machines

A. For the notion of a digital state machine
   1. Minsky, Marvin, Chapters 1 & 2 of *Finite & Infinite Machines*, Englewood Cliffs, N.J.: Prentice-Hall (1967), pp. 1–31.
      - A very simple reminder of the notion of a discrete (digital) finite-state automaton. Included as background.

B. For the notion of digitality
   1. Haugeland, John, Chapter 2 ("Automatic Formal Systems") of *Artificial Intelligence: The Very Idea*, Cambridge, MA: The MIT Press/A Bradford Book (1985), pp. 47–84.
      - A somewhat more extensive development of Haugeland's notion of digitality than that in "Semantic Engines" (#1). *AI: The Very Idea* is a great conceptual introduction to artificial intelligence; it is the book I recommend to outsiders to convey a sense of what AI is up to (accessible to the paradigmatic reader of the *New York Review of Books*, for example). It may be less clear here than in "Semantic Engines" that Haugeland is not a believer in AI.
   2. Goodman, Nelson, Chapter 4 ("The Theory of Notation") of *Languages of Art*, Indianapolis and New York: Bobbs-Merrill (1968), pp. 127–173.
      - A very useful analysis and set of distinctions relevant to the analog/digital debate. It is a little tough to read through Goodman's nominalism (and sometimes also through his ego, though neither is as distracting in this chapter as in some others in the book), but well worth the effort. Goodman is no computer scientist, but the analysis is both more careful than many others, and also very applicable to the computational case. Note in particular how he applies the basic notion of discreteness twice — once at the syntactic level, once at the semantic level — in contrast to Haugeland, who provides only a single-level analysis (this stems from Haugeland's idiosyncratic treatment of formality; see the comment at #1, above).
   3. Lewis, David, "Analog and Digital," *Nous*, **V**:3 (Sept. 1971), pp. 321–327.
      - A short, clear paper, that pretends in part to be a critique of Goodman, but is in fact an analysis of a very different set of intuitions. Where Goodman is concerned with the difference between discreteness and continuity, Lewis attempts instead to reconstruct the difference between analog and digitial computers. The paper is useful in part simply because it is instructive to see the two distinctions part company, quite apart from whether one ends up agreeing with Lewis' talk of "multi-digitality."
   4. Haugeland, John, "Analog and Analog," *Philosophical Topics* (Spring 1981); reprinted in J. I. Biro & Robert W. Shahan, (eds), *Mind, Brain, and Function: Essays in the Philosophy of Mind*, Norman, Oklahoma: Univeristy of Oklahoma Press (1982), pp. 213–225.
      - A terrific short paper. Not just a summary of Haugeland's own analysis of digitality,

which we have seen elsewhere (#s 1 and 11), but a useful reply to and critique of both Goodman (#12) and Lewis (#13). The real significance, however, comes in the last few pages, in Haugeland's discussion of second-order versions of digital and analog — something that I think is an extraordinarily important idea.

5. Dretske, Fred I., "Sensation and Perception," Chapter 6 of *Knowledge and the Flow of Information*, Cambridge, MA: The MIT Press/A Bradford Book (1981).
   - As Dretske himself admits, a use of "familiar terminology—analog vs. digital—in a slightly unorthodox way." Important, though, because it points to some of the consequences of digital representation — particularly the sense or intuition that when you have extracted digitally encoded information you have in some sense "bottomed out" in terms of wringing everything from the representation that is there to be wrung. Dretske also makes the remarkable claim that the fundamental difference between perception and cognition is that perception deals with information encoded in analog form, whereas cognition deals with it digitally. The boundary between perception and cognition, therefore, is according to Dretske the process of converting from analog to digital representation. As usual, although the discussion is not conducted in computational terms, the morals are relevant to the computational case.
6. Fodor, Jerry A., and Ned J. Block, "Cognition & the Analog/Digital Distinction," unpublished ms.
   - In this unpublished manuscript (of mediocre print quality — sorry about that) Fodor and Block disagree not only with Lewis, but also with any intuitive connection between the analog/digital distinction and the continuous/discrete distinction. Instead, they tie their notion of analog to whether a machine's state transitions are lawlike, under the appropriate description.

V. Information Processing (to be skipped)

A. For the syntactic notion

1. Weaver, Warren, "Recent Contributions to the Mathematical Theory of Communication", in Claude E. Shannon and Warren Weaver, *The Mathematical Theory of Communication*, Urbana, Illinois: The University of Illinois Press (1949), pp. 1–28.
   - From its title, and from the fact that it was written a year after Shannon's original work, you might expect this paper to present further results that had appeared since that paper's publication. But you would be wrong. Instead, this is Weaver's conceptual interpretation of Shannon's results, plus his (Weaver's) argument that they were more semantically important than Shannon himself thought. Clear, historically important, and particularly interesting with regards to the semantic developments of #s 20–24, below, which of course did not appear until more than 30 years later.
2. Shannon, Claude E., Introduction and Part I (pp. 29–64) of "The Mathematical Theory of Communication", in Shannon & Weaver (1949), pp. 29–125.
   - Given the Weaver paper (#17) and Singh's introduction (#19), I have included this primarily for historical interest. This is the paper (originally a Bell Labs technical report) that spawned the whole field of so-called (syntactic) information theory. We will not be concerned with any of the mathematical details, so Part I should be looked at only to see what Shannon is up to conceptually. Note in particular his explicit comment (2nd ¶

of the introduction) that "semantic aspects … are irrelevant to the engineering problem," in spite of what Weaver argues in his §3.

3. Singh, Jagjit, Chapters 1–9 of *Great Ideas in Information Theory, Language, and Cybernetics*, New York: Dover (1966), pp. 1–126.
   - Since logic, formal symbol manipulation, automata theory, and recursion theory (to say nothing of many more practical aspects of computer science) are all more likely to be familiar to students in the class than is classical information theory, I have included the first third of this book. Partly is will serve as background (review or introduction), and partly it will convey a sense of how notions from the syntactic version of information theory were thought not only to be applicable to, but in fact to explain, notions of both analog and digital computers (chapters 8 and 9, respectively). A clear introduction. Includes a chapter (7) on the tie to physical notions of entropy, a forerunner of recent work at the junction of the foundations of computability theory and quantum mechanics.

B. For the semantic notion

1. Dretske, Fred I., "Précis of Knowledge and the Flow of Information," The Behavioral and Brain Sciences, 6 (1983), pp. 55–91.
   - A synopsis of Dretske's seminal book, which shifted attention away from a pure focus on the syntactic details of encoding, and brought genuine semantical issues of content back into legitimate theoretical view. Dretske has since admitted that the explicit linkage to the Shannon-Weaver theory at the beginning is less relevant than he himself thought at the time he wrote it. This paper is important because it lays the ground work for the Barwise-Perry-Israel work on information (#21), which in turn influenced both Rosenschein (#23) and Halpern (#24).
2. Dretske, Fred I., Chapter 3 of *Knowledge and the Flow of Information*, Cambridge, MA: The MIT Press/A Bradford Book (1981).
   - The crucial chapter from Dretske's book, laying out the essentials of his semantic notion of information. Whereas the précis (#20) conveys a sense of Dretske's overall project, this chapter is a better presentation of the details of the underlying intuition of correlated correspondence.
3. Israel, David and Perry, John, "What is Information?", in Philip P. Hanson, ed., *Information, Language, and Cognition*, Vancouver: University of British Columbia Press (1990), pp. 1–19. Also available as CSLI Technical Report CSLI–90–145.
   - Summarises the notion of information that Barwise and Perry first discuss in their *Situations and Attitudes* (Cambridge, MA: The MIT Press/A Bradford Book, 1983). As with Dretske, this is a semantic notion (thus compare this paper with Singh's chapter 2 of the same name!). Note the relativisation of information to constraints, the attempt to work out some actual details of how information is structured, and the tantalising final sentence adverting to the potential use of representation (especially striking given the non-representational emphasis of *S&A*). Does this constitute agreement, in the end, with Fodor's formality condition (#7)?

C. For application of the semantic notion to AI and computer science (respectively)

1. Rosenschein, Stanley J., "Formal Theories of Knowledge in AI and Robotics", *New Generation Computing*, 3 (1985), pp. 345–357. Also available as CSLI Technical Report No. CSLI–87–84, Stanford University.

- The paper in which Rosenschein introduces his notion of "situated automata," defined with respect to a semantic notion of information (which he calls "knowledge") along the lines of #s 20–22, above. The discussion is framed in AI terms, but the approach is applicable to computational systems more generally, as Stan himself now advocates. Within AI circles, the situated automata approach is often compared with that of such other non-representationalists as Chapman and Brooks, though Rosenschein (as his title suggests) is much more of a formalist than they are.

2. Halpern, Joseph, "Using Reasoning about Knowledge to Analyze Distributed Systems," *Annual Reviews of Computer Science*, volume 2 (1987), pp. 37–68.
   - Although the machinery used in this paper is derived from logics of knowledge and belief, the fundamental notion is very close to that of semantic information (counterfactual-supporting correlation) discussed in #s 20–22. Halpern surveys its use for analysing problems in distributed computation. As usual, our present interest is not in the mathematical details, but in the conceptual structure of his project — as illustrated for example in ??????, where he takes up the question of the relation between knowledge and representation.

VI. Some Applications to Practice

1. Scott, Dana, and Strachey, Christopher, "Toward a Mathematical Semantics for Computer Languages", *Proceedings of the Symposium on Computers and Automata*, Polytechnic Institute of Brooklyn (1971), pp. 19–46. Also available as Technical Monograph PRG–6, Oxford University Computing Laboratory.
   - An early and very clear introduction to the enterprise of giving mathematical semantics to programming languages, known in computer science under the label "denotational semantics." R. D. Tennent has a similar tutorial in the *Communications of the ACM* (August 1976, 19, pp. 437–453), but the original Scott & Strachey paper is a paragon of conceptual clarity, for example in keeping use and mention straight. Do not worry about technical details; focus instead on exactly what the project actually comes to.
2. Barwise, Jon, "Mathematical Proofs of Computer System Correctness", CSLI Report No. CSLI–89–136, Stanford University.
   - A review of, and contribution to, a spirited debate several years ago about the validity and viability of "proving programs correct". This debate, and the underlying issues at stake in it, provide a good chance to review some of the conclusions we have wrestled with in this course: about what computation is, how it relates to the world, and what aspects are reconstructed by different kinds of analyses.
3. Smith, Brian Cantwell, "The Correspondence Continuum", CSLI Technical Report No. CSLI–87–71, Stanford University.
   - Although written primarily with the knowledge representation community in mind, this paper attempts to clarify some of the issues of what I have sometimes called "computation in the wild" (see especially pp. 4–6, for example, which separate programs from processes), especially with regard to the various semantical issues that come up in the computational case.

——end of file ——