Frames,Brains,and Chinese Rooms:
Problems in Artificial Intelligence

A THESIS SUBMITTED TO
THE FACULTY OF THE LIBERTY UNIVERSITY
SCHOOL OF RELIGION
IN PARTIAL COMPLETION OF THE DEGREE
MASTER OF ARTS
DEPARTMENT OF PHILOSOPHY

by
JEFFREY KOPERSKI

LYNCHBURG, VIRGINIA
SPRING, 1991
ABSTRACT

Advocates of strong artificial intelligence believe that properly programmed computers can go beyond the simulation of intelligent acts so as to instantiate and exhibit true intelligence, that is, intelligence equivalent to that of man. In this thesis, I consider three problems for strong-AI.

First, John Searle’s well-known thought experiment of the “Chinese Room” is used to reestablish the syntax-semantics distinction and to show how this distinction applies to computer programs. I review the Chinese Room, consider a variety of objections to it, and then expand on the key points in Searle’s work.

Second, I examine the Frame Problem in artificial intelligence, a question made popular by Daniel Dennett. Rational agents have the ability to adjust their conceptual schemes and update their noetic web of beliefs so as to maintain a representation of the world. This ability is easily observed but not well understood. I argue that computers lack this ability altogether. The Frame Problem examines this deficiency and programming techniques designed to overcome it.

Third, the Overseer problem examines the need for artificial systems to have a rational agent in place who designates a given task and determines when that task is successfully completed by the system. I argue that as long as this need exists, artificial systems cannot be considered intelligent in an unequivocal sense.
# TABLE OF CONTENTS

## ABSTRACT

INTRODUCTION

I. The Problem

II. General Outline

III. An Important Qualification

NOTES

CHAPTER 1: SYNTAX, SEMANTICS, AND SEARLE

I. Searle's Chinese Room: The Argument

II. The Chinese Room: Application

III. Facing the Critics

A. "Causal Powers"

B. Counterexample: Haugeland's Demon

C. The Systems Reply

D. Faulty Model of the Mind-Brain

E. Misleading Analogy

F. Intuition Pump

IV. Beyond the Chinese Room

A. Another Thought Experiment

B. Trying to Fix RR

C. Operationalism

V. Epistemology and Attribution

VI. Conclusion

NOTES

CHAPTER 2: THE FRAME PROBLEM

I. Background Knowledge

II. The Frame Problem(s)

A. Data Retrieval

B. Relevant Facts

C. Selective Updating: The Bookkeeping Problem

III. Repairs and Solutions

A. Repairs

B. Scripts: A REAL solution?

IV. Future Hurdles

A. The Folding Problem

B. The Jumping Problem

V. A Real Problem?

VI. Conclusion

NOTES

CHAPTER 3: THE OVERSEER PROBLEM

I. Connectionism: The New Frontier

II. The Overseer Problem

A. The Overseer and PDP

B. The Overseer and GOFAI

C. Objection: PDP Can Overcome OP

III. Induction

IV. Thesis Conclusion

NOTES

SELECTED BIBLIOGRAPHY
INTRODUCTION

We are about to conceive of the knower as a computing machine.
Warren S. McCulloch, M.D.
"Through the Den of the Metaphysician," 1954

I am prepared to go so far as to say that within a few years, if there remain any philosophers who are not familiar with some of the main developments in artificial intelligence, it will be fair to accuse them of professional incompetence, and that to teach courses in philosophy of mind, epistemology . . . without discussing . . . aspects of artificial intelligence will be as irresponsible as giving a course in physics which includes no quantum theory.

Aaron Sloman
The Computer Revolution in Philosophy, 1978

Often in the history of philosophy, philosophers have had to field concerns, questions, and claims from other disciplines. This is such a time. Artificial intelligence, once an obscure corner of computer science, raises questions and makes claims that traditionally belong to the epistemologist and metaphysician, to the philosopher of language and mind, as well as the psychologist and neurophysiologist. In this work, I will examine some of the philosophical concerns surrounding these claims.

I. The Problem

Let's start to outline the field of interest with some terms. The first is "strong artificial intelligence." This term names the position that, as Michael Arbib says, "AI programs really could exhibit understanding or intelligence, rather than simply simulate aspects of behaviors we construe as intelligent when performed by a human being" (emphasis mine). "Weak AI" is similar to the former view although weak AI is limited to simulation and makes no claims about true machine intelligence. I will use "AIers" to designate...
supporters of strong-AI. "AI" itself is an elusive term to define, but here are a couple of attempts by those in the field:

*AI is that part of computer science concerned with designing intelligent computer systems, i.e. systems that exhibit the characteristics which we associate with intelligence in human behaviour---e.g. understanding language, learning, reasoning, solving problems etc.*

[A system is artificially intelligent] if the task [the system] is carrying out would require intelligence if performed by humans.

Why is it that AI has been such a hot topic philosophically? One reason is that philosophers perceive a major misunderstanding of computer science claims at the popular level. This often happens when technical terms---in any field, not just computer science---are imported back into popular writing. Another reason lies in the sometimes outlandish claims of AIers that reach far beyond their field. Consider for example this scenario envisioned by Carnegie-Mellon researcher Hans Moravec. One day all knowledge and skills will be stored in computers and "down-loaded" to people whenever they require such expertise.

This will result in a gradual erosion of individuality, and formation of an incredibly potent community... [Which will be] constantly improving and extending itself, spreading outwards from the solar system, converting non-life into mind... This process, possibly occurring now elsewhere, might convert the entire universe into an extended thinking entity.

If a philosopher or theologian made such claims, he would not be taken very seriously.

Let's examine strong-AI more closely. "Intelligent
"behavior" is picked out, somewhat intuitively, and identified with those acts carried out by humans that artifacts (normally) cannot carry out. Computers can be made to act intelligently via rule-governed symbol manipulation (i.e., they are programmed). The digital computer is a systematic symbol shuffler. That is, at one level of computational description, 1's and 0's are moved to the right place at the right time according to the hardware restraints and the software instruction. At a deeper level, transistors are systematically turned on and off. The program determines the rules by which these manipulations will occur. According to this "rules-and-representations" view (RR), humans also must or might (there is a spectrum of opinion here) produce intelligent behavior through an internal set of rules. Supporters of RR point to logic and language as paradigm cases of rule governedness that would require such mental symbols. Both humans and computers would, therefore, comprise a larger class of "information processing systems." Although the RR view does not apply to all branches of computer science, it has dominated the AI community for the past twenty years. Strong AI working under the RR model is now called classic or "good-old-fashioned-AI" (GOFAI, coined by John Haugeland).

Traditionally optimism runs high in GOFAI. Herbert Simon of Carnegie-Mellon University claims that literally thinking machines now exist. In fact, these machines have
thoughts in the strictest sense of the term. Alan Newell, Simon’s long-time associate, claims that intelligence just is physical symbol manipulation. If machines manipulate the symbols in the “right way,” the machines should embody intelligence in precisely the same sense as humans. In a paper by both Simon and Newell they write,

There are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in the visible future—the range of problems they can handle will be coextensive with the range to which the human mind has been applied.

Marvin Minsky of MIT, perhaps the most important thinker in AI, thinks that humans might be reduced to household pets by the next generation of computers. Finally, John McCarthy, inventor of the term "artificial intelligence," says the "ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer operating systems. . . ." For example, your furnace thermostat has at least three beliefs: it’s too hot, it’s too cold, it’s just right. These are a few of the more celebrated opinions found in GOFAI circles.

II. General Outline

The subject matter discussed so far is both broad and deep enough to take in many directions. I will focus on three, one per chapter.

Chapter 1 will deal with the syntax-semantics distinction—actually the failure to maintain this distinction. AI critics argue that at the core of any
digital computer system is a network of switches. The on-off manipulations (even the 1's and 0's in the machine code) are purely syntactic: tokens are shuffled around in a rule-governed way. Alers counter that at the system level (i.e., the system-as-a-whole), complex rule-governed behavior allows for the emergence of semantic content: the tokens will have meaning. John Searle's "Chinese Room" thought-experiment will serve as the pivotal example through which to address this problem. I will argue, in agreement with Searle, that syntax is never sufficient for semantics, although the Chinese Room example is not without its shortcomings.

Chapter 2 will focus on what some believe to be an insurmountable epistemic problem for the RR view, viz., the Frame Problem. In short, the frame problem is the computer's inability to abstract (what we consider to be) important details from common experience and to then use this information to guide its future behavior. I will argue that the frame problem is not another technical difficulty to be overcome in time, but rather that the classic RR model is inadequate to handle this difficulty. AI is quite possibly in the midst of a paradigm shift because of the chronic failure to overcome this problem and because of the early successes of a rival model.

As Thomas Kuhn has shown, to have a paradigm shift, there must be a rival paradigm. In Chapter 3 I will briefly present some key aspects of the rival "connectionist"
approach. The main topic of the chapter is to present yet another difficulty, what I will call the Overseer Problem, that affects not only GOFAI computers, but the new connectionist systems as well.

III. An Important Qualification

Equivocating over technical-operational and popular terms has a long and glorious history in AI. Some in the artificial intelligentsia purposefully deny that there is any equivocation. McCarthy provides a key example:

To ascribe certain "beliefs", "knowledge", "free will", "intentions", "consciousness", "abilities" or "wants" to a machine or computer program is legitimate when such an ascription expresses the same information about the machine that it expresses about a person.\(^1\)

Here is a clear case where the metaphorical use of anthropomorphic terms has ceased to be metaphorical. I see nothing wrong with using mental terms to describe the behavior and function of various computer operations; however, we must realize that such ascriptions are (usually) intended as analogies only. For example, one might describe a chess-playing computer as believing that its king is in trouble. In fact, the computer "believes" nothing nor does it have any conception of "king" or "chess" for that matter. The computer is simply executing the commands it has been programmed to carry out. Mental terms used in this analogous way are (or at least were) useful shorthands in denoting a given behavior.

In Section I of his paper "Artificial Intelligence
Meets Natural Stupidity," Drew McDermott explains that programmers sometimes become entranced by their own "wishful mnemonics." McDermott shows how a suggestive subroutine like UNDERSTAND, GOAL, or ASSERT might get its name before the programmer knows if his algorithm has any chance of simulating what's being named.

If [the programmer] calls the main loop of his program 'UNDERSTAND', he is (until proven innocent) merely begging the question. He may mislead a lot of people, most prominently himself, and enrage a lot of others. What he should do instead is refer to this main loop as '60034' and see if he can convince himself or anyone else that 60034 implements some part of understanding.

Such oversights ultimately hurt the AI field. Slogans and buzz words, especially when used outside of a technical context, have come to confuse more than clarify.

With this danger plainly in view, such metaphorical terms still have heuristic value. As long as the metaphor is clearly noted, using "learns," "sees," etc., to describe computer behavior is a useful shorthand. Thus the reader is warned up front that my use of anthropomorphic terms in this manner in no way endorses a reduction of any kind. With this qualification firmly established, let's enter the Chinese Room.
NOTES


3Ibid.


By 'behavior' here and throughout this work I mean the movement of parts by the entity in question.


5By 'behavior' here and throughout this work I mean the movement of parts by the entity in question.


6Ibid.


Searle, 30.


11Ibid.


13Ibid., 144.
CHAPTER 1
SYNTAX, SEMANTICS, AND SEARLE

For Searle, intentionality is rather like a wonderful substance secreted by the brain the way the pancreas secretes insulin. Brains produce intentionality, he says, whereas other objects, such as computer programs, do not. . . . Luckily for us . . . our brains make intentionality; if they didn't we'd behave just as we now do, but of course we wouldn't mean it!

Daniel Dennett
"The Milk of Human Intentionality," 1980

There is a tendency in AI today towards flashy, splashy domains. . . . Yet there is no program that has common sense; no program that learns things that it has not been explicitly been taught how to learn . . .

Douglas Hofstadter
"Artificial Intelligence: Subcognition as Computation," 1983

In the twentieth century, the philosophy of language has become a hotly contested sub-discipline. Regardless of one's position, all agree that many helpful distinctions have been made. The distinction I will focus on in this chapter causes little controversy until computer intelligence comes into play. This is the distinction between syntax and semantics.

Syntax is the realm of symbols (tokens) and their proper manipulation. 1 Syntactic rules tell us "what counts": which tokens are allowed and in what order they are to be recognized. For example, among the specified tokens in arithmetic we have `1', `2', `3', `+', and `=' . In first grade we learn that `2 + 1 = 3' is an allowable string of arithmetic tokens whereas `2 1 + = 3' is not. The symbols are arbitrary, of course, e.g., we know that `1 + 8 = 9' is the same as `I + VIII = IX' in Roman numerals. But what do
we mean by "is the same as?"

To answer this question, we enter the realm of semantics. The Arabic and Roman numerals are the same in the sense that we assign the same semantic content to '8', 'VIII', and 'eight'. Tokens themselves lack semantic content; they are meaningless. Tokens only have meaning when they are interpreted.

Computers are syntactic engines. In the introduction, we saw that they manipulate symbols according to the rules of a given program. Alers (recall this designates strong-AI, RR enthusiasts) claim that syntactic rules can be of sufficient complexity when embodied in a computer program so that a computer can go beyond simple syntax and actually understand the commands being executed. That is, given a sufficiently rich syntactic program code, the computer will provide semantic content to its symbols. Under the RR model of cognition, we do much the same thing—people have an internal syntax that produces our semantics.

John Searle's main criticism of this view is simple:

There is a distinction between manipulating the syntactical elements of language and actually understanding the language at a semantic level. What is lost in the AI simulation of cognitive behaviour is the distinction between syntax and semantics.

Searle's thought-experiment illustrates that manipulating symbols according to a list of rules will never produce understanding in the mechanism—biological or mechanical—executing the rules. In short, syntax is never sufficient
I. Searle's Chinese Room: The Argument

The experiment runs as follows (I will put myself in Searle's role). Imagine I am in a small room with baskets full of Chinese language characters. It is important to note that I do not understand Chinese in the least—the characters are just different tokens to me. I have been provided with a rule book, written in English (which I obviously understand), with instructions for matching these symbols with other symbols. I use the term 'symbol' here under the assumption that these tokens symbolize something to someone. The rules in the book govern the manipulation of the symbols by their shape alone; no translations or meanings are provided.

Outside the room, native Chinese people have access to two slots, In and Out, and have no idea what's going on inside. They slip pieces of paper with Chinese characters through the In slot and I match these symbols with those specified by the rule book. The book instructs which symbols to then pick out of the baskets and send through the Out slot.

Unbeknownst to me, the people outside are putting questions, not just random symbols, in the In slot and, from their point of view, answers to these questions are coming through the Out slot. This exchange is exactly what the writers of the rule book intended. These "answers" to "questions" are in perfect Chinese syntax and obey standard Chinese semantic rules. As far as the Chinese are concerned,
whoever is in the room seems to understand their language. But I do not understand Chinese.

The analogy to a digital computer lines up this way. The rule book corresponds to a computer program written in whatever artificial language you like (e.g., LISP, the traditional language for AI). The rule book writers are the programmers, the only one's who know what's going on inside and outside the room. The baskets of symbols are a data base. I am the computer, or perhaps specifically the CPU. The In slot serves as an input terminal and the Out slot is a printer.

The thrust of the argument is really quite simple:

If I do not understand Chinese solely on the basis of running a computer program for understanding Chinese, then neither does any other digital computer solely on that basis. Digital computers merely manipulate formal symbols according to rules in the program."

We may break the argument down this way:

α is able to execute a list of rules that simulates the understanding of language L to some observer B.
α does not understand L.
Therefore the execution of a list of rules is not sufficient for α to understand L.

II. The Chinese Room: Application

The thought-experiment is intriguing and for the most part uncontroversial as it stands. Of course, Searle uses the Chinese Room as a launching pad for a more elaborate argument consisting specifically of four premises (P) and four conclusions (C).»

(P1)Brains cause minds. This premise is unnecessary
and serves primarily to affirm Searle's physicalism. The mind is viewed here as a higher order property of the brain (higher, that is, than say its greyness or solidity). Searle might agree with Minsky's belief that "minds are what brains do"; however, Searle strives to preserve the concept of mind from hasty reductions, especially those of the eliminative materialist. Except for this premise, as Sir John Eccles points out, Searle's argument could easily come from a dualist in the philosophy of mind.

(P2) Syntax is not sufficient for semantics. Of course, strong-AI calls this begging the question. The RR model claims precisely that semantics will be found to be a property of a sufficiently rich syntax. The Chinese Room is meant to emphasize (and perhaps reestablish) P2.

(P3) Computer programs are purely formal (syntactic). There is no fundamental reason for programs to be run only on digital computers since the program itself only specifies syntactic manipulations. The symbols at the machine code level (1's and 0's, on and off) are wholly abstract and can be assigned to any rule-governed system: water pipes, control relays, or, according to Searle, "old beer cans strung together with wires and powered by windmills." Keep P3 in mind; it will become important later on.

(P4) Minds have mental contents; specifically, they have semantic contents. This premise Searle takes as self-evident and all of cognitive science assumes it. The cause,
not the fact, of semantic content is controversial. Searle now moves to his conclusions.

(C1) No computer program by itself is sufficient to give a system a mind. Programs, in short, are not minds and they are not by themselves sufficient for causing minds. The only things we know that cause minds, at this point, are brains. Some mental activities (e.g., logic) can obviously be simulated by computers, so in a (trivial?) sense minds can be described computationally. However, simulated "thinking" via computational symbol manipulation does not imply that thinking is identical to symbol manipulation.

(C2) The way that brain functions cause minds cannot be solely in virtue of running a computer program (P1 + C1). This strikes at heart of the RR model. No matter what kind of internal syntax the RR might claim we have, it will never be rich enough to explain our semantic behavior.

(C3) Anything else that caused minds would have to have causal powers at least equivalent to those of the brain. The opacity of the term 'causal powers' will attract critics en masse. This problem will be examined in the next section.

(C4) For any artifact that we might build that had mental states equivalent to human mental states, the implementation of a computer program would not by itself be sufficient for those mental states. Searle does not deny the possibility of synthetic intelligence, just that such intelligence will not simply implement a formal program.
III. Facing the Critics

In the last ten years, Searle has become one of the most prominent philosopher-critics of GOFAI. Not surprisingly, the Chinese Room has drawn fire from all corners of cognitive science. In this section, I will examine some of these criticisms and offer possible replies. I assume that Searle would agree with these replies to his critics, since some of these he offers himself, but I cannot guarantee this across the board.

A. "Causal Powers" [C3]: What does this mean? Many critics wonder aloud what these causal powers might be. Searle leaves his explanation at a rather intuitive level saying only that physical systems do not exhibit intentionality—at least not yet. However, as a physicalist, Searle cannot call on a mind or soul to serve as the metaphysical seat of either agent causality or intentions. Other physicalists want a material analysis of these causal powers, and rightly so.

Reply. Whatever they are, the brain's causal powers are more than just the ability to execute the next rule in a syntactic code, which is all the symbols in a program can do.
Without question, we do need science to help uncover the nature of these causal powers of the brain, but the causal powers of 1's and 0's are already well understood: they are abstract symbol carriers only. Furthermore, whether a machine has these causal powers is an empirical question. How so? Recall Searle has no theoretical objections to the possibility of synthetic intelligence (see C4), thus machines with causal powers might be invented someday. The point is, instantiating a program is not sufficient to provide such machines with causal powers. It is this syntactic insufficiency that the Chinese Room is directed against, not the possibility of man-made intelligence.

I agree with Searle's critics that "causal power" is a very opaque term that invites abuse. If such powers are empirically testable, as Searle claims, it would help if he provided at least an outline for the criteria we might employ to find them. Without any empirical guidelines, such causal powers remain quite mysterious and out of place for a physicalist. In fact, intentionality and agent causation are traditionally called on by dualists to criticize reductive theories in the philosophy of mind. Critics rightly sense that these elements are difficult to make coherent in a physicalist system.

B. Counterexample: Haugeland's Demon (H-demon). In trying to nail down Searle's application of causal powers, John Haugeland has proposed this counterexample. Consider
person α who has been struck with a rare disease such that his brain's neurotransmitters no longer send signals from neuron to neuron. We install in α's brain an H-demon that "tickles the appropriate synapse of the next neuron in a way that is functionally indistinguishable, to that neuron, from the arrival of genuine neurotransmitters." The demon is so quick that it never falls behind and α's brain continues to function. The question for Searle is, does this brain still have intentionality (and therefore causal powers)?

Reply. Searle does not back down: "[α's] neurons still have the right causal powers; they just need help from the demon." That is, if the H-demon can reproduce the activity within the brain as if it were working properly, then yes, the causal powers are maintained and so is intentionality.

Searle's critics do not like this reply since it only slightly firms up the notion of causal powers. The empirical criterion asked for seems to be simply "x has causal powers iff x is a working brain." I believe, however, that Searle's reply is consistent with his main point against GOFAI. Note that the H-demon does not follow a set of rules (e.g., a computer program) to keep α's brain going. This new demon-brain system might be semi-artificial or synthetic (thus possibly intentional, see last reply), but as long as the demon-brain operates by a means other than rule execution, the H-demon is not an example of strong-AI. Therefore this
is not a counterexample to the Chinese Room.

C. The Systems Reply. This criticism comes from both computer and neuro-science. In short it says, "You do not understand Chinese, but the room as a whole does." The Chinese Room is not complete if we just focus on me, the guy inside. The room is also data banks of symbols plus the rule book plus scratch pads. . . . Understanding is ascribed to the system, not just me. As Haugeland puts it, "the system as a whole manipulates the tokens in ways appropriate to what they mean, with no intervention from outside users; and that's semantic activity." Searle confuses different levels of description and attribution. According to Daniel Dennett, for example, "I understand English; [my] brain doesn't. . . ." No one wants to ascribe understanding to the computer's CPU. Searle is simply looking too deep. If the system exhibits semantic activity, then at the system level we can ascribe understanding to it.

Reply. System level ascription does not escape the syntactic-semantic distinction: "[If] I, as the central processing unit, have no way of figuring out what any of these symbols mean; . . . [then] neither does the whole system." That is, if I don't understand the Chinese symbols, then no matter how many useful things you throw in with me, the room doesn't understand either.

Consider a counterexample, suggested by Searle. I memorize the rule book and the symbols in the baskets. The
system is no more; there’s just me. I can now apparently read and write Chinese, but the question is whether I understand Chinese the same way I understand English. Clearly I do not. I am following the rules I have memorized for dealing with Chinese characters, but I still don’t know what they mean, "Whereas the English subsystem knows that 'hamburger' refers to hamburgers, the Chinese subsystem knows only that 'squiggle squiggle' is followed by 'squoggle squoggle'."

The key difference again rests on what I, as a knowing subject, self-reflectively understand or do not understand. No outside observer can unerringly make this judgment.

D. Faulty Model of the Mind-Brain. Although this criticism has many versions, I will focus on Marvin Minsky’s:

I don’t mean to say that brains or minds are simple; brains are immensely complex machines—and so are what they do. . . . Whenever we speak about a mind, we’re referring to the processes that move our brains from state to state.16

Minsky’s point is closely related to the usual criticisms about folk psychology. That is, most or all of the events named by prescientific mental terms, including Searle’s "intentionality" and "causal powers," are simply the result of a highly complex, physical process. Eventually these prescientific terms will be reduced or replaced by a mature mind science. Searle’s analysis forces us to give credence to opaque terms derived from our own limited introspection. Before we say what computers can or cannot understand, let’s
first allow neuroscience and AI to run their course so we really know what's going on.

**Reply.** Searle's reply continues on the same theme. A fully mature neurophysiology with precise scientific jargon will not erase the difference between a belief ascribed to an artifact and a belief had by a person. For the sake of scientific research, one can certainly put humans, computers, and thermostats on a relative "belief continuum." But the point Searle comes back to again and again is that our ascriptions cannot change the real, first-person, qualitative difference between a machine's behavior and our experience. The gulf between my knowing Chinese and behaving as if I know Chinese cannot be bridged by a new scientific description of the process.

**E. Misleading Analogy.** The Chinese Room gives the impression that the subject is doing the equivalent of manipulating an AI program by hand. The reader is led to identify with Searle's "feeling the lack of understanding Chinese." But human execution of a complex language program made for a digital computer would take years! Searle doesn't mention this and for good reason. If the question-and-answer transaction with the Chinese people outside the room took years, the questioners would no longer believe anyone in the room understood Chinese. Once this time factor is revealed, Searle's intuitive link with the reader fails and so does the thought-experiment.
Reply. The responses to this problem are concise and compelling. First, we could replace the man in the room with Haugeland's H-demon, which is very fast. Hand manipulating the program now is on par with a computer. But when we ask the (English speaking) H-demon if he understands Chinese, we get the same result as before. Second, when did speed become a criterion for intelligence? How fast someone solves a problem or thinks about a question might be a measure of his intelligence, but time is not a factor in determining whether a subject is intelligent. Bringing in a speed element is ad hoc.

F. Intuition pump. For those already sympathetic to the strong-AI side, this is possibly the most damaging criticism. Unfortunately for Searle, his entire "argument" is nothing but an intuition pump. He doesn't really assert anything, but Searle gets the reader to nod his head in agreement and say to himself, "Yea, he's right. I would surely know whether I understood Chinese are not." According to Ned Block, Searle has a hidden premise: "Evidence isn't sufficient to overrule the intuitions." After all, our intuitions once told us Earth is stationary, large objects fall faster than small, etc. When evidence is introduced, our intuitions must conform. Searle is unfair in two ways. First, he acts like evidence is at the mercy of intuition. Second, he does not deal with any of the evidence for the AI side. Such hand-waving over the successes of the field
should not be tolerated.

Reply. 'Intuition' is used here in an equivocal way. On one hand, one's intuitions (i.e., thoughts or opinions) about the goings on in the Chinese Room are irrelevant. The point of the illustration is to emphasize a conceptual truth that is usually well understood: syntax is not semantics. The Chinese Room attempts to recapture the idea that shuffling uninterpreted formal symbols is not the same as understanding their meaning. On the other hand, 'intuition' has an epistemic sense regarding one's first-person experience of a situation. I know intuitively (i.e., directly and with certainty) that I don't understand Chinese. This use is different from the popular use of 'intuitive insight' as in "women's intuition." The second use of 'intuition' is what Searle appeals to in most of his replies. No third-person observer can have a subject's direct, first-person intuition that the subject does or does not understand a given symbol.

In conclusion, Searle's critics often try to take the Chinese Room beyond the very limited scope intended. In almost every case, the reply refocuses the discussion on the key issues of syntax-semantics and first-person understanding (or lack of understanding). These are the conceptual pillars Searle is trying to save from a behavioral reduction.

IV. Beyond the Chinese Room

Why is it that the Chinese Room invokes so much
criticism? Most Aiers point to the intuition pump. The computer science successes brought about by the RR model are thought to be a decisive blow against anyone (especially bothersome philosophers) who relies solely on non-empirical arguments. Although complete brain simulation might be a technological impossibility, under the RR theory there is no physical impossibility preventing the brain’s rule-governed activity from being captured by a program. If we could only discover the correct rules and if we had a medium of sufficient complexity, Aiers claim, we would have unequivocal, non-metaphorical, artificial intelligence.

In this section, I will present an adaptation of the Chinese Room to try to determine where the line is drawn on the application of the RR model (i.e., what is it that we may call intelligent?). RR enthusiasts will likely object that my application is not what they intended their criteria to be used for; however, the thought-experiment is a well established device to determine where a theory might lead.

A. Another Thought Experiment. As I was studying one morning, a small robot crawled into my room, handed me a book, and promptly disappeared. The book (copyright 2025) contained the history of AI research. In the year 2020, engineers at IBM-Xerox, Inc. invented (i.e., will invent) the UBS (Ultimate Brain Simulation) program. UBS was run on a highly advanced optic-digital computer, the CRAY-7. With speed and memory many orders of magnitude higher than in
previous decades, UBS on the CRAY-7 can produce any semantic behavior desired. According to the now mature RR model of cognition, the CRAY-7 instantiates true intelligence.

In 2021, Ed, an electrical engineer who dabbled in necromancy, got a copy of UBS but decided not to run it on the CRAY-7. A formal program, after all, has no intrinsic preference for what physical apparatus carries it out.

In Ed's wizardly experiments, he learned to conjure up a small demon (much like the H-demon described earlier). This H-demon is not only quick but transdimensional, "blinking" in and out of any location instantaneously. Ed teaches the H-demon to leave flashlights at every planet and moon in the galaxy. Ed decides to run UBS on a galactic scale.

Impossible? As I mentioned in the introduction, at a deep level of description a digital computer is a system of electronic switches that are either on or off, depending on the program instruction. Theoretically, any medium that can keep track of two physical states (on and off) can carry out a program.

Ed teaches the demon to read the UBS code and switch the flashlights on-off as the code instructs. The demon periodically checks with Ed for any new commands ("interrupts" in computer jargon).

Now if running UBS instantiates true human-level intelligence, should we expect the CRAY-7, and now the galaxy
itself, to have beliefs, perhaps an idea of "self," and even free will? What criteria would we apply? The philosophically interesting question the galactic brain experiment points to is, if intelligence is a property, I, what are we to count as candidates for intelligence, x? Since persons (a) are intelligent—the paradigm case if you will—certainly the variable x can be replaced with the name of any person, Ia (read "a is intelligent"). AIers also want to substitute certain computers (c) that meet the RR criteria for intelligence, Ic. Now we have another medium that meets the RR criteria; but does anyone want to count the galaxy (g) as a candidate for intelligence, Ig? What criterion in the RR model allows Ia and Ic but disallows Ig? There are none.

B. Trying to Fix RR. There are two possible criteria to limit the domain of x. First, AIers could argue the galactic brain is too slow: Ix only if x is fast. This rule is derived from the fact that semantic behavior is not observable from a system that takes months or years to reply to an inquiry. The problem with this criterion is that the RR model does not consider speed. Adding a time qualifier is clearly an *ad hoc* fix, but let's allow it for now.

The galactic brain could still meet the new criteria. Instead of one H-demon blinking around space, Ed conjures up trillions of H-demons. Better still, Ed discovers that demons have trans-dimensional sight. He puts one H-demon per
flashlight so each demon can read the program code on Earth and also see the other lights going on/off. Speed would then approach the new requirements.

The second new criterion possible is spatial localization: $I_x$ only if $x$ is spatially localized. $g$ is undeniably non-local, but this fix fails as well. What is the standard for localization? On a human scale, $g$ is not localized; on a universal scale, $g$ is very localized (i.e., $g$ is not undeniably non-local). Like the speed objection, this one is another ad hoc fix. I cannot alter the galactic brain to fit this time, but such subjective standards tend to weaken the strong-AI case.

I believe the "what counts" problem for the object of predicate $I$ runs throughout the Chinese Room debate. Consider three levels of use for mental terms, including intelligence. Level one ($L_1$) is the neural level. Many believe $L_1$ is where we find the sufficient conditions for semantic behavior. Neuro-science focuses here. $L_2$ is the personal level: the common use of mental terms has to do with people, not neural interactions. Psychology focuses on $L_2$ while AI picks and chooses between $L_1$ and $L_2$. There is also an $L_3$ at the level of corporate objects like nations and companies. We say things like "General Motors loves its customers" and "the United States wants all foreign governments. . . ."

Pure reductionists want to dispel with $L_2$ and $L_3$. 
Prereflectively, everyone agrees that attributing mental properties or actions to corporate objects is highly metaphorical. Reductionists go further to say mentalistic terms used in folk psychology are likewise non-technical and confusing and will eventually be replaced by scientific terms (see Minsky above). We should disregard the upper levels of mental description in favor of what's "really" going on: neural interaction.

The more common view is to agree with the reductionist about corporate objects, but reject the wholesale reduction of L2 to L1. Mental terms like 'belief' and 'intention' are not sufficiently captured or explained by neural phenomena. This is the position of Searle and substance dualists.²²

C. Operationalism. There is a shadowy middle ground between reductionism and folk psychology that sometimes goes unrecognized in the AI debate. This territory is held by operationalism, an approach to the philosophies of mind and science that will become very important in chapter 3. The operationalists hold that L2 terms (beliefs, desires, etc.) are useful and should not be reduced to L1 terms. This differs with Searle and the dualists in that, although useful, L2 terms for the operationalist have no more ontological weight than L3, the corporate-mental terms.

Here is an example by Dennett. Consider terms used by loggers in Maine:

You can 'trick' an apple tree into 'thinking it's spring' by building a small fire under its branches in the late fall; it will
blossom. This way of talking is not just picturesque and is not really superstitious at all; it is simply an efficient way of making sense of, controlling, predicting, and explaining the behavior of these plants in a way that nicely circumvents one's ignorance of the controlling mechanisms.\(^2\)

L2 is especially needed when a system becomes too complex to predict its behavior in L1 terms. That is, it's fine to talk about a robot "wanting to go outside" (L2) when an explanation in terms of program code and electronics is too long or complex. 'Wanting' conveys the idea adequately.

An alternative use of mental terms is seen in the programmer who starts with L2 and works his way to L1. In a chess program, for example, the programmer wants to make the computer protect-the-king. The notion of protect-the-king must then be translated down to the L1 program code level to make the system perform the L2 behavior. L2 is not a property of the system, rather L2 reflects our stance or attitude toward the system.\(^2\) Note that operationalists in AI are usually operational only with respect to L2 and L3 and scientific realists regarding L1. L2 has heuristic, predictive value only and has nothing to do with the nature of the system under investigation--man or machine. L1 is where real explanations are found.

Why is operationalism important here? I mention this approach because operationalists use mental terms much like Searle or a dualist would use them. For the latter two, however, L2 has more than just heuristic value. L2 for Searle says something about the system itself, not just how
we might view it. The AI debate is about whether a system really is intelligent or has desires. Operationalism does little to help answer this question, but might sound to the unwary reader as if it does.

V. Epistemology and Attribution

Some of the confusion over the Chinese Room could be eliminated if all the participants would maintain a consistent epistemic relationship to the system being discussed. That is, critics often alter their epistemic position without warning the reader. By 'epistemic position' (EP) I mean the relationship between an agent and a state of affairs whereby the agent is justified in holding certain beliefs about that state of affairs.

I'm sure an example would be useful. Say I am in my study and I come to the belief $p$ that my wife is doing cartwheels in the living room. Now I have no evidence for this, thus $p$ is unjustified given my epistemic position (which coincidentally corresponds to my physical location). If however, I go into the hall and see my wife's shadow flipping all about and hear the crash of a lamp on the floor, my epistemic position has changed. I now have some justification for $p$. It is important to note that epistemic position does not affect the truth of any belief. I can be justified in a belief and still be wrong. This last point is simply the well-known distinction between truth and justification.
Let's apply EP to the Chinese Room. The Chinese people only have access to the IN/OUT slots of the room. Call their epistemic position EP1. Given the apparent replies they receive to their questions, those at EP1 are justified in believing $p_1 = \text{'someone inside the room understands Chinese'}. $

Those in the strong-AI camp argue that we are in an analogous EP to those at EP1 when we approach a highly sophisticated computer system. That is, given the system's semantic behavior, we are justified in believing $p_2 = \text{'the computer understands } z \text{'}$ where $z$ is the subject matter of the program (e.g., chess). Just like the Chinese people, we have behavioral evidence that the computer understands what it is doing. $p_2$ might be false, but given our EP, we may rightly attribute intelligence to certain intricate systems. Or so the argument goes.

Let's go back to the Chinese Room. Say one of the Chinese doesn't like the answer he received and takes a sledge hammer to the OUT slot. The window breaks revealing Searle with his rule book and baskets of symbols. As the Chinese begin to investigate, they come to realize that Searle doesn't understand a bit of Chinese; he's just doing what his rule book tells him. Now their EP has changed from EP1 to EP2 given the new evidence. At EP2, they are no longer justified in believing $p_1$.

The key to Searle's experiment is that we are not in an analogous epistemic position to EP1 with respect to complex
computers. Rather, we are closer to EP2. We know exactly what's going on inside: rule-governed symbol manipulation following the instructions of a well-defined computer code. We can get beyond the behavior of the system down to how the (apparent) semantic activity has been generated. 

An anticipated reply to this analysis of EP comes from the classic "other minds" problem in philosophy. Searle briefly considers this himself. I can't know with certainty that anyone else has a mind (or consciousness or intelligence, etc.) like I know that I have a mind. Everyone else could be an android controlled by Descartes' evil genius. From my EP, all the evidence available to me for believing you have a mind is your behavior. Why not use the same behavioral criteria for a computer?

In reply, note that the AI criticism short-changes us on the available evidence. The criteria I use to infer that others have minds is based on our similar behavior in similar situations, but behavior is not the only thing we have in common. There are both behavioral and physical similarities to consider, specifically, I and other humans share a close neurophysiological similarity that I do not share with computers. This commonality is the backbone of Richard Swinburne's principles of charity and credulity used to deal with the other-minds objection (see his Evolution of the Soul pp. 13-17 for a detailed analysis). The problem with attributing a mind or intelligence to a computer is our
obvious lack of similarity in physiology. At present, neuroscience is not mature enough to say how far the notion of intelligence can be removed from the only paradigm case available: human brains. Swinburne points out that,

It would be different if we had a well-justified general theory of consciousness ... that explained which physical processes of kinds currently unknown give rise to which mental events ... Then we could examine the Martians and robots to see whether their physical processes were of a character to give rise to mental events, i.e. were similar to our own in whatever respects the theory had identified as crucial for this. 27

In lieu of such a comprehensive mind-theory, our EP allows us to conclude only that beings with both behavior and physiology like ourselves are intelligent. In terms of the earlier discussion, Ix is limited to x's that are human, at least for the time being. 26 If behavior were all we had to go on to judge whether a given computer is intelligent, then our EP would allow for the attribution of intelligence. But we are not so limited.

VI. Conclusion

Neither Swinburne (a dualist) nor Searle (a physicalist) objects to theoretical synthetic intelligence. The point of the Chinese Room is that whatever form this synthetic intelligence takes, it will not simply instantiate formal rules. It must affix semantic content as well. The driving cognitive model of AI, the RR model, is inadequate for true intelligence, since it maintains that syntactic rules are
sufficient for semantic behavior. But as we have repeatedly seen, there is a distinct, first-person, qualitative difference between my understanding of, say, English, and the ability to act as if I understood English by following a rich syntactic rule book.

The galactic brain thought-experiment and the discussion on epistemic position have demonstrated the difference between our attribution of intelligence to systems for heuristic purposes, and the question of whether a system is intelligent or not. This distinction must be kept in mind when reading the vast array of literature on the Chinese Room and AI itself.
NOTES

1 The term 'symbol' is somewhat ambiguous. One usage is such that something can only be a symbol if it is viewed as symbolic of something else. The notion of symbolism, something standing for something else, is semantic. The usage here is syntactic, referring to the figure itself: '׳', '®', '¥', '±', '¢' . . .


3 The Chinese Room is discussed in several publications by Searle. The best known is "Minds, Brains, and Programs," The Behavioural and Brain Sciences 3 (1980): 417-457. This includes numerous commentaries representing all parts of cognitive science.


10 Ibid.

1 Searle, "Minds, Brains, and Programs," 452.


Searle, Minds, Brains, and Science, 34.

Searle, "Minds, Brains, and Programs," 419.


Searle, "Minds, Brains, and Programs," 452.

Hofstadter and Dennett, 373.

Term used by Dennett, 429.


For readers with computer backgrounds, Ed designates specific flashlights to serve as internal and external memory. Flashlights in our solar system would serve as one register and likewise for other systems. Whole systems would have normal hexadecimal addresses. The H-demon is really just a super, serial data-bus. It carries the appropriate on-off signals to flip-flops (flashlights).

To be precise, Searle believes mental terms are not yet captured by full physical explanations but should be eventually. These explanations will not, however, support the RR model. Dualists, on the other hand, hold that mental phenomena is not fully reducible to physical phenomena even in principle.


See the "combination reply" in Searle, "Minds, Brains, and Programs," 421. Searle’s argument is along the lines of EP, although he does not use these terms.

Ibid., 421-422.

Anticipating the outcry from my theistic readers, I realize I have limited $x$ so that God is likewise not a candidate for $Ix$. Please note that the predicate $I$ is to be taken in a restricted sense along the lines of intelligent-like-ourselves. God is certainly intelligent in the common sense of the term, but he is not intelligent-like-ourselves. This is clearly seen in God's omniscience, which is something quite different from intelligent-like-ourselves.
CHAPTER 2
THE FRAME PROBLEM

Within a generation the problem of creating artificial intelligence will be substantially solved.

Marvin Minsky
Computations: Finite and Infinite Machines, 1967

The AI problem is one of the hardest science has ever undertaken.

Marvin Minsky
"How Can Computers Get Common Sense?" 1982

In Chapter 1 we saw how the rules-and-representations model of cognition used by GOFAI (good-old-fashioned-AI) failed to distinguish computer syntax from semantics. In this chapter we will examine another problem that has plagued GOFAI for two decades: the frame problem (hereafter FP).

Before I say what the problem is, the reader should be warned about some side issues. First, there are many articles in print that show how some writer mistakenly identifies the "real" FP and then goes on to clear things up. Unfortunately, these articles do not always agree on what the real FP is. I will not try to sort out this confusion nor will I adopt any one writer's choice of terms. 'FP' here will represent the most general rubric for several related and over-lapping topics one of which might be called the frame problem proper.

Second, the FP is related to but not the same as a "frame," "script," or "schemata." These terms, now common in the AI literature, refer to a programming strategy employed to help overcome the FP. This particular strategy will be
discussed at the end of the chapter.

The FP is a question about the relation of propositions to one another. Persons, we observe, make key adjustments to their noetic web of propositions without being told to do so. All the sub-topics within the FP, to be discussed in Section II, relate to the digital computer's inability to either draw on or properly manipulate propositions the system already contains.¹ I will not offer a solution to the FP here. My goal is to demonstrate a rational ability we have that computers lack. This lack emphasizes the gap between man and machine that strong-AI claims to have bridged.

I. Setting the Stage: Background Knowledge

Before jumping into the subtopics that collectively make up the FP, I would like to introduce a related difficulty. This information should help the reader gain a foothold in the discussion. Perhaps the easiest way to introduce the problem of background knowledge is with my favorite illustration by Daniel Dennett, the man perhaps most responsible for bringing the FP to the attention of the AI community. Consider the "snack problem."

 fences. I couldn't make a sandwich without knowing a good deal—about bread, spreading mayonnaise, opening the fridge, the friction and inertia that will keep the turkey between the bread slices and the bread on the plate as I carry the plate over the table beside my easy chair... I listed a few of the very many humdrum facts one needs to know to solve the snack problem, but I didn't mean to suggest that those facts are stored in me—or in any agent—piecemeal, in the form of a long list of sentences explicitly declaring each of these facts for the benefit of the agent... We know trillions of things; we know that mayonnaise doesn't dissolve knives on contact, that a slice of bread is smaller than Mount Everest, that opening the
Rational agents have the ability to form, not just call up from memory, the specific propositions required to solve everyday problems. As Dennett points out, there are many propositions that we "know" but do not directly consider as we go about our day.

The problem of background knowledge arises due to the program requirements of serial computers. To solve any problem, a computer must be given a data base of all propositions needed to solve that problem (i.e., the initial conditions) plus instructions about how these propositions relate. The difficulty lies in programming the computer to use its data base appropriately. Of course, the notion of 'appropriate' is quite vague. Somehow we draw on our experience to make inferences that relate to our present situation. A computer must be instructed (i.e., programmed) to make similar right inferences—"right" being another ill-defined notion in this context.

One major difficulty in this project is that we are not sure ourselves what principles we use to learn from experience. Margaret Boden, a prominent AI writer-philosopher, believes that in all human reasoning there are unformalized "integrative principles of tacit inference or global knowledge of which one is not introspectively aware." Although the nature of these principles is of great interest,
we do not need to fully understand how we use our experience to in fact make use of it.

This pragmatic use-it-without-understanding-it approach is fine for humans. Nonetheless, whatever these principles are that we take advantage of, computers do not have them—thus the problem. Once again, all initial conditions plus the principles of inference between propositions must be provided for the computer. Since we do not know ourselves how we in fact make use of experience, it is no surprise that formalizing this ability into computer code is a formidable task.

How past experience affects future events is of course not a new puzzle. The problem of induction remains unsolved with no solution on the horizon. The background knowledge problem might simply be a byproduct of the problem of induction, but, as Dennett rightly points out, the FP as a whole will remain even if induction is resolved. I will come back to this later. Let's now move on to a variety of topics that all claim to be at least part of the FP.

II. The Frame Problem(s)

The FP in its broad sense is a computer's inability to "know" how any single piece of information affects the rest of a database. Let's say for simplicity's sake that a database, at a certain level of description, contains propositions. When a programmer inputs a new proposition, how does a computer determine which other propositions are
affected? There is currently no way to determine which specific propositions are to be changed without an exhaustive search of the data base. Such a search, however, is highly inefficient and not computationally practical given the time restraints we put on intelligence (i.e., systems that take days to solve simple problems are not considered candidates for intelligence even by strong-AI).

To illustrate, say I have a large stack of index cards, with one sentence per card, that collectively explains the United States strategy for arms negotiations with the Soviet Union. Intelligence sources then report that the Soviets have secretly withdrawn all troops from Poland. This new information will require a revision in some of the sentences on my index cards, but how many and which ones? The only way to know is to look through them all.

Dennett sees the FP not as a technical hurdle for AI, but a general epistemic question: How does any cognitive creature know which conceptual propositions need to be updated to keep one’s internal model "roughly faithful to the world." I will of course limit the discussion to the AI realm and not explore the larger epistemic question.

For the most part, this section will only present the difficulties for AI. How programmers do in fact circumvent (or at least minimize) the FP will be dealt with in Section III. Let’s now examine the components of this large problem.

A. Data Retrieval. Assuming that all the requisite
facts for solving a given range of questions are provided in a data base, how does a computer get to these facts to make use of them? Well-known AI critic Hubert Dreyfus explains,

To establish that a fact exists in its data banks a computer must retrieve it. Worse, to establish that some fact is not in the data bank requires examining the entire list of what the computer knows to determine that the fact in question is missing.

In contrast, Dreyfus cites Richard Shaffer's example of our (usually) direct access to our knowledge. I know immediately when I was born and with some thought I can recall when my mother was born. I know immediately that I do not know when Thomas Jefferson was born and no amount of thinking will retrieve that information. I know that I do not know it. In a computer, only an exhaustive search can reveal the absence of any fact.

If the reader does not think exhaustive searching is much of a hindrance, especially with the speeds at which modern computers operate, consider that the world's best chess playing system, Carnegie-Mellon's "Deep Thought," can calculate 750,000 positions per second. By 1992, the rate should exceed one billion positions per second. For the system to calculate the best move in any given situation, an exhaustive search of all possible counter-moves and counter-counter-moves, etc., would have to be made. However, it is estimated that there are $10^{120}$ different possible games of chess. Even if Deep Thought could calculate a billion games per second, an exhaustive search would take over 100 trillion
trillion centuries. This will not do. How is it that the chess program on my PC thrashes me in much less time?

The answer is that programmers are well aware of algorithms for more efficient data base searching. One such method is the use of heuristics. Heuristic rules are short-cuts or rules-of-thumb that people often use in decision making. For example, in chess I use rules-of-thumb like "don't sacrifice a bishop to capture a knight." Rules like this are a common programming tool to help cut down computation time.

There are two major problems with the heuristic solution. First, the approach makes the problem less noticeable but fails to solve it. For a data base search that is both accurate and fast, very specific heuristics are required. Such rules are often hard to specify when the data becomes overly large and complex. The second objection is more problematic: the rules don't always apply. As any chess player knows, sometimes you have to sacrifice the bishop. The second is a more formidable problem since the computer has no way of determining when such rule breaking is allowed unless there is yet another rule to tell it to do so. That is, the system would need second order heuristics for breaking first order heuristics, and so on. A point of diminishing returns develops such that the time spent searching for applicable rules sacrifices the time saved by employing heuristics in the first place.
To digress for a moment, this need for rule breaking points to what some call the hard/soft paradox of AI. Some human reasoning appears to be rule-governed (e.g., logic, grammar, etc.). The computer’s algorithmic ("hard") rules are perfectly suited to simulate such thinking. Hard rules cannot be violated except by the instruction of another hard rule.

In everyday situations, however, we find many instances where rules are appropriately broken. Consider the rule "do not throw cold water on your wife." This rule should immediately be broken if my wife’s hair catches fire. The "soft" side of human reasoning is the ability to intuit circumstances that call for extreme or unusual action. In these cases, the computer’s algorithmic rules become a hindrance.

I can think of few rules of behavior that persons should not violate under any condition. I do not believe, however, that I have a ready made criteria for identifying the circumstances under which such rule breaking is required. Such decisions must often be made "on the spot." This need for adaptability is a notorious problem for AI.

In sum, although programming techniques such as heuristic rules lessen the data retrieval problem, no method thus far has solved it. There appears to be a profound difference between men and digital computers regarding memory itself and the relation between memory and behavior.
B. Relevant Facts. Perhaps the key difficulty in the FP is determining the relevancy of facts. That is, given the vast number of facts available to make any single decision, how can a computer choose the relevant ones and ignore the rest?

Consider again my stack of index cards on arms negotiations. Someone asks "if the Soviets destroy half of their ICBM's, how many cruise missiles can we give up?" I would like to consider only those cards that have to do with cruise missiles, but I don't know which ones they are until I search through the whole stack skimming for the word "cruise." Furthermore, there may be some cards that do not have "cruise" on them, but are relevant to the question. How do I get to these without wading through the entire stack again? It appears that the exhaustive search continues to be the only way to be sure.

Again heuristic rules can be used to help determine relevancy. Consider Haugeland's theoretical computer with an English language data base that must determine how to interpret 'the pig is in the pen'.10 'Pen' of course has multiple uses in English. The computer must determine if 'pen' is a place on a farm or a writing instrument. The relevant fact for solving the ambiguity is that pigs cannot fit into a writing instrument. The problem is, how does the computer determine the relevant fact in this case has to do with size? How does the computer determine the key feature
in any case?

A possible solution to the Haugeland example is that "in" usually has a size implication. The computer could solve the ambiguity with the heuristic semantic rule "a sentence of the form 'x is in y' implies that y is larger than x." The drawback to this solution is twofold. First, as the number of semantic rules like this one becomes large, the computer would need meta-semantic rules to determine which semantic rules are relevant. How will the computer know if any of these rules (semantic, meta-semantic, and so on) should be broken? More rules for rule breaking are required. Once again, heuristics push the problem back but do not solve it. Second, "in" is itself ambiguous in this context. This particular semantic rule will not work for the sentence "The pig is in the photograph." Other non-heuristic methods for determining relevant fact will be presented in the next section.

Before moving on, let me refer back to the problem of induction. Following Dennett, let's assume a computer has somehow overcome the problem and now has perfect inductive "beliefs." The computer still suffers from the FP since it will still have no way of prioritizing this massive array of beliefs about the future. Exhaustive knowledge about the future, based on past experience, does not insure that such knowledge will be used effectively.

A walking encyclopedia will walk over a cliff, for all its knowledge of cliffs and the effects of gravity, unless it is
designed in such a fashion that it can find the right bits of knowledge at the right times, so it can plan its engagements with the real world (emphasis mine).

C. Selective Updating: The Bookkeeping Problem. Some consider the bookkeeping problem to be the key difficulty of the FP. Consider a database of propositions that collectively form a model of, say, a desk with colored blocks on it: Blockworld. Let the model be output in three dimensional graphics so everyone can see what Blockworld looks like. Blockworld is set up to correspond to a group of real colored blocks on my desk. Thus when I turn the blue block on my desk on its side, I input a new proposition, \( p \), to the Blockworld model, \( p = \text{The blue block is turned on its side} \).

The question is, which other propositions need to be revised when \( p \) is added to keep the model accurate? Of course all of the propositions which contain 'blue block' might need revision and only those need be considered if the blue block is alone in the middle of the desk. But what if ten other blocks are stacked on the blue block? Now a large number of spatial propositions need to be revised that do not contain 'blue block'.

Depending on the overall situation, some propositions must be updated and others left alone. Provided that causal interactions are all part of the model (e.g. moving blue block causes ten others to fall), the computer must access each proposition in the model to find if it needs revision.
But as we saw, exhaustive searches are time consuming and heuristics alone do not solve the problem. An efficient method for selectively updating only the relevant information is required. Of course this assumes the relevant facts problem has already been solved.

Let's now look at some of the programming techniques used to circumvent the FP.

III. Repairs and Solutions

A. Repairs. To lessen the effects of the FP, some applications use the "cheap test." The program contains commands that exclude irrelevant parts of the data base from the search. For example, in Blockworld there might be a limiting rule like 'when an object moves, color and size are not affected'. The data base could be easily organized so that propositions about color and size are grouped. The program would then "flag" these groups to exclude them from the search.

Unfortunately, the cheap test is actually a kind of heuristic and is subject to the drawbacks mentioned earlier. Furthermore, such limiting rules do not always apply. If the objects in Blockworld are ice cubes or clay, for example, friction will change the shape of the blocks when they are moved. By now the reader should see the relationship between the scope of application and the proportional need for more (perhaps second order) rules.

Another programming technique is the "sleeping dog"
approach. The program is instructed to leave all propositions alone unless there is some positive reason to revise them. That is, when a new proposition is introduced, the program assumes that without sufficient warrant or computational relevance nothing else in the data base is affected.

The glaring difficulty with using the sleeping dog approach in any general application is how to specify "sufficient warrant" or "computational relevance." Defining these terms and then encoding them for the computer is the FP! I do not mean to imply that either the cheap test or the sleeping dog approach are useless. They are quite sufficient for dealing with a variety of applications. The criticisms here are intended to show that neither is able to solve the FP for very general or complex applications.

As Dennett has pointed out, there appears to be another inherent problem with all such relevancy tests. The goal is to make the system limit its focus to only certain (i.e., "relevant") inferences. The two approaches mentioned add a relevancy axiom to cut down on the calculated inferences. Dennett points out, however, that in any (monotonic) deductive system, the addition of an axiom never reduces the number of inferences available, it always increases them. Otherwise, the new axiom must be inconsistent with a previous one.

For example, consider a closed deductive system with
five axioms. Let's say that there are twenty proofs calculable from these axioms. Now add a sixth axiom. If Axiom 6 makes any of the previous twenty proofs invalid, it is inconsistent with one of the first five axioms. Say Axiom 6 is consistent with the first five and is a relevancy test which is supposed to limit the number of inferences. Instead of limiting steps, the program will calculate all the same inferences (the twenty proofs) plus calculate their relevancy (i.e., solve the proofs that follow from the addition of Axiom 6, the relevancy test).

What we really want is for the system to ignore irrelevant data, not calculate that the data is irrelevant and then ignore it. No one wants a computer to waste valuable computation time calculating all of the propositions that it can ignore.

B. Scripts: A REAL solution? The most successful solution to the FP to date is the use of "scripts" (Schank), "frames" (Minsky), and "schemas" (Rumelhart). For those unfamiliar with field, the difference is negligible. The dissimilarity lies in the degree of a program's "anticipation." Here are the computer scientists' own explanations of this approach:

Minsky: A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed... Much of the phenomenological power of the theory hinges on the inclusion of
We define a script as a predetermined causal chain of conceptualizations that describe the normal sequence of things in a familiar situation. Thus there is a restaurant script, a birthday-party script, a football game script, a classroom script, and so on.

The ability-to-ignore, lacking in other techniques, is not attained through the addition of a new IGNORE-algorithm in the program. Instead the system’s attention is focused by the stereotypical expectations of the script.

This anticipatory behavior of a script roughly simulates our own day-to-day interactions. When someone enters a familiar situation, like Schank’s football game script, he has certain expectations and customary actions that help him to socially negotiate the activity. If he encounters something highly unusual, say his chili-dog bursts into a chorus of "God Save the Queen," it takes time to sort out what’s going on and what the appropriate reactions might be.

Similarly a robot running scripted software has no difficulties within the preprogrammed expectations. That is, when the robot’s encounters fit the script, its reactions are easily accessed by the program thus circumventing exhaustive searches, relevancy tests, etc. Abnormal encounters take longer to deal with, not unlike reactions in persons.

There are some usual questions a GOFAI critique asks at this point. The first of which usually involves adaptability: Does the script technique allow for a wide
range of applications? When things proceed as usual, the computer's script can deal with most problems and has an acceptable range of adaptability. I will not defend this assertion except to say computer scientists would not continue to pursue such a research program without limited success. When things come "out of the blue" however, like the aberrant chili-dog, the computer's reactions are often not foreseen nor acceptable.

A more severe problem is related to the relevancy test. Not only do persons determine which facts are relevant in a given scenario, but they also assign different degrees of relevancy to them. We are able to adapt to different "levels of weirdness" as John Searle puts it. But even scripted software is not able to prioritize its expectations to suit different situations. For example, in Schank's restaurant script, "it is equally 'weird' for the restaurant to be out of food as it is for the customer to respond by devouring the chef." The singing chili-dog is just as strange in the football script as kicking a seventy yard field goal.

How is it that we recognize degrees of weirdness? At least part of the answer is that we are not isolated to facts within a given script; we have access to other facts about cultural norms and interpersonal relations. For example, a woman without a top walking on a beach is highly irregular (i.e., the situation, not the woman) in America but not in France. An isolated group of "beach facts" cannot be
assigned degrees of abnormality without knowledge of cultural norms. Notice how the relevancy test, degrees of relevancy, and background knowledge problems all come into play here.

Given these problems with the script technique, why has it been such a successful research program? The answer ironically yields another criticism: there is an ad hoc fix available within each script. As a programmer debugs a given script, he usually stumbles onto the abnormalities that go beyond the software's ability to adapt. At these specific points, a direct contingency command is implanted to prevent the system from "locking up" or doing whatever undesired activity it tends to fall into when confronted with aberrant data. Individual scripts can always be saved in this way.

For example, consider a simple algebraic computer program written in BASIC in which some variable $A$ is used in several equations. One of the lines of the program is $500 \frac{(B+C)}{A}$. The programmer notices that if the variable $A$ is zero, then line 500 will generate an error message: division by zero is algebraically undefined. To circumvent this problem, the programmer can put in a line '490 If $A=0$, then 550' which instructs the program to skip over line 500 and execute line 550 if $A$ is zero. This strategy is a perfectly acceptable ad hoc fix for preventing division by zero, but it is obviously restricted to this problem.

Unfortunately, ad hoc solutions are not sufficient to solve the FP in general. Scripts are a useful approach for
solving specific problems but this strategy cannot overcome
the digital computer's chronic lack of adaptability.

Scripted software was hoped to be the key to solving a
host of GOFAI puzzles. In the early 80's it became evident
that current scripts were not performing as expected and some
of the strong-AI rhetoric started to be toned down. For
example, compare Minsky's 1967 quote at the beginning of the
chapter with this one from 1981:

Just constructing a knowledge base is a major intellectual
research problem... We still know far too little about the
contents and structure of common-sense knowledge. A 'minimal'
common-sense system must 'know' something about cause-and-
effect, time, purpose, locality, process, and types of
knowledge... We need a serious epistemological research
effort in this area.²⁰

Let's assume that scripts in the future will overcome
all the aspects of the FP mentioned so far. There appear to
be two more problems on the horizon that affect this
approach.

IV. Future Hurdles

A. The Folding Problem. Assume for the moment that
humans work from something like scripts. As I sit here in
the midst of my thesis-script, say I come to a sticking
point—I just don't know how to work out some conceptual
problem. But I recall from my history-class-script that in
World War II the Americans, in order to liberate others,
initially bypassed well fortified islands held by the
Japanese. I take the "principle" (an admittedly ill defined
notion that I will not shore up here) of temporarily
bypassing difficult obstacles and apply it within my thesis-script by moving on to the next topic. In script terminology, I have folded information from two unrelated scripts. Programming a computer to do likewise is the heart of the "folding problem."

Within the limited universe of a given script, some AI programs adapt well to new information and can generally limit the crunch of the FP. Some AIers believe that when enough powerful scripts are loaded into one computer, it will behave intelligently in all script-scenarios and therefore, under GOFAI, will be intelligent.

The folding problem is simply this: computers do not channel general principles between isolated scripts. As my previous illustration shows, we are able to learn from a given situation and apply our knowledge to new unrelated settings. All inferential ties between software scripts, in contrast, must be determined in advance.

Furthermore, there are also instances when whole, unrelated scripts need to be integrated. Assuming persons rely on a restaurant-script and a birthday-party-script, we can easily fold these two scripts together when a birthday party is held in a restaurant. Such smooth combinations of discrete scripts do not just emerge within a program. To solve the folding problem, programmers must find a way to tie together the conceptual archipelago of multiple scripts.

B. The Jumping Problem. Closely related to the folding
problem, the jumping problem is the computer's inability to make a smooth transition between scripts.

For example, say you are eating lunch at the Western Steer (i.e., are in the middle of your restaurant-script) when your colleague, who has illegally made his third trip to the one-time-only salad bar, begins choking on a tomato. You must immediately transition from your restaurant-script to a choking-script to save his life. An observing computer meandering through its restaurant-script has no ready made way of jumping to another. Inference bridges would have to be provided in advance between all possibly connected scripts—obviously demanding a great deal of foresight on the part of the programmer.

Scripts are unquestionably useful within a well defined scenario. The folding and jumping problems show, however, that daily cognitive activities require interaction between normal, stereotypical situations. In the next section I will examine some criticisms directed at the FP itself. That is, some claim the FP is a pseudo-problem.

V. A Real Problem?

The most outspoken antagonist of the FP is Drew McDermott. His criticism is three-pronged. First, the above mentioned sleeping dog method has been a sufficient programming technique for most of the last fifteen years. It is so successful, in fact, that no one in the field is even working on a solution to this mythical FP! Second, regarding
intelligence, we do not make the same demands of perfection on humans that the FP imposes on computers. That is, humans cannot take all relevant information into account when such facts become numerous. Information overload is a problem for all cognitive beings, not just computers. Third, the philosopher-critics ("framist") who push the FP are a moving target. Once AIers begin to answer the real FP, framists shift to other "related" problems that are themselves able to be overcome.

Patrick Hayes, in the article right after McDermott’s in one anthology, responds to McDermott’s challenge.

The frame problem is sometimes dismissed as being a narrow, technical problem of little philosophical interest... I think this is a mistake. For one thing, a "narrow technical problem" which is this immediate, this central, this devastating, and this resistant to solution is worthy of some respect.22

Why isn’t anyone working on the FP? Because, as I pointed out earlier, for each script under consideration, there is always an ad hoc fix that takes care of that and only that script. Whatever this fix may be, it generally is not applicable to another script.

The overload problem in McDermott’s second prong is for the most part correct. Although he fails to consider our abilities to prioritize incoming information based on experience, McDermott rightly points out that GOFAI critics should not demand that a computer surpass man’s cognitive abilities. All finite beings can consider only a finite number of propositions at one time and are thus subject to
information overload. McDermott's point is granted, but unfortunately for GOFAI, the FP is more than just a breakdown of information management.

Another reason for taking the FP seriously, albeit a somewhat anecdotal one, is that it appears to be at the heart of the "conversion" of one of GOFAI's key workers, Terry Winograd. According to Dreyfus, Winograd now teaches Heidegger to his computer science students at Stanford to help show the difficulties of formalizing background knowledge and making scripts interact. The point is, McDermott is simply wrong when he claims that computer scientists do not feel the tension of the FP. Winograd is the most visible strong-AI "defector" to date.

VI. Conclusion

How does this chapter support my overall case against strong-AI? I believe the problem helps show that human knowers are different from (at least digital) computers in kind, not simply in degree of complexity. The rational, human ability to make key changes to one's noetic web without rules or instructions on how to make such changes is not shared by program driven systems. The FP shows that simply having more powerful programs/rules does not eliminate the problem. This unique rational ability is evidence that men are more that very fast, very powerful, and very complex rule-governed machines. As long as this man-machine gap remains in place, the claims of strong-AI will retain their
status as optimistic exaggerations.

To conclude, I would like to digress into epistemology proper for a moment. In most epistemic models, except very pure forms of coherence justification, philosophers realize that some beliefs play a more significant role in our noetic structure than others. These "weightier" beliefs support the lesser ones or at least require a greater amount of evidence before they may be revised. How beliefs relate to and rely on one another is a subject of great debate (e.g., what is it for one proposition to be evidence for another?). Understanding the nature of this relation is not required for persons to, in fact, hold and prioritize their beliefs.

This lack of knowledge about knowledge will not do for GOFAI. Programmers must guess how inference, evidence, and even induction work and then go on to formalize these opaque notions. In this light, it is little wonder why philosopher's often view GOFAI claims with skepticism.
NOTES

1 Recall the anthropomorphic qualifier in my introduction. I will use "proposition" in a nominalist sense to make it applicable to computers. This move is not meant to imply any bias in philosophy of language or metaphysical controversies.


3 My computer literate readers are already wondering whether parallel (as opposed to serial) systems will solve the upcoming problem. The answer is 'no,' parallelism is a treatment, not a cure for the FP. Massively parallel systems, however, are radically different. I believe the so-called "connectionist" architecture takes care of most of the FP, but not the overseer problem, the subject of Chapter 3.


7 Ibid.


11 Dennett, "Cognitive Wheels," 140.
12Haugeland, 83.

13Ibid., 84.

14Dennett, "Cognitive Wheels," 142.


17Dennett, "Cognitive Wheels," 144.


19Dennett, "Cognitive Wheels," 144.


21Dreyfus and Dreyfus, "How to Stop Worrying about the Frame Problem Even though It's Computationally Insoluble," 107.


23Hayes, 126.

CHAPTER 3

THE OVERSEER PROBLEM

If we are to avoid the morass of metaphysics (1), we must reduce as many concepts as possible to numerical terms. On the other hand, we must face the fact that the most important aspects of human life are intrinsically non-numerical. Any attempt to ignore this is highly unscientific. In the true intellectual approach, one accepts this fact and copes with it.

Richard Bellman
Polytechnic Institute of Brooklyn Symposium on the Mathematical Theory of Automata, 1962

The conceptual difficulties discussed in the last two chapters usually involve serial digital computers. On the cutting edge of computer technology lies another architecture that may very well overcome both the challenge of the Chinese Room and the Frame Problem. The programmes are often called connectionism, parallel distributed processing (PDP), neural networks, and massively parallel systems, although these names are not fully synonymous. There is unfortunately no space here to provide an adequate overview of this new approach, however, there are a handful of introductory articles available.¹

In this chapter I will present another problem for strong-AI that affects both conventional computers and PDP. I must ask the reader to assume that what I attribute to PDP is correct and not open for discussion at this point. The purpose here is not to quibble about what PDP does or how it accomplishes its tasks; I will simply grant most of the claims made by computer scientists in this young field. I will then go on to show that neither digital computers nor
PDP's escape what I call the Overseer Problem (OP).

I. Connectionism: The New Frontier

Recall my Blockworld illustration from Chapter 2. In short, the computer's program constructs an internal model of a room with colored blocks. The key to Blockworld is the software: the better the program, the better the model. In contrast, PDP does not use a program. Instead the system is trained (invoking the anthropomorphic qualifier one last time) through examples. That is, the system develops its own generalizations and internal representations from particular examples without algorithmic instructions on how this representation is to be constructed.

For example, let's say a given neural net receives input from a video camera that is pointed at various live dogs. After a large number of sample-dogs has been given to the net, the system will form an internal representation of a paradigm-dog. Once trained (not programmed) the net can distinguish dogs from other objects. Giving a sophisticated net a great deal of training should allow it to distinguish dogs from cats, cows, and perhaps even from statues of dogs.

For the reader whose knowledge of computers is limited to the digital variety, it might be difficult at first to appreciate the tremendous difference between a conventional and a non-programmable system. The key is that no program means no rules; PDP rejects the RR theory of cognition.

The approach that we take in developing PDP models is completely different [from serial digital computers]. First, we
do not assume that the goal of learning is the formulation of explicit rules. Rather, we assume it is the acquisition of connection strengths which allow a network of simple units to act as though it knew the rules.²

PDP might offer a way around the Chinese Room since, without a program, there is no (prescriptive) syntax.³ Few are claiming that neural nets are sophisticated enough to "understand" a natural language; however, the syntax-semantics distinction that the Chinese Room relies on is harder to define when there is no program. As for the Frame Problem, PDP memory is not stored in data-bases but rather throughout the net (cf. holographic images). For reasons that I cannot explore here, such content-based memory makes the relevant facts issue much less of a problem. Whether PDP can solve the Frame Problem or the Chinese Room must wait for another time. Let's now examine the problem PDP does not escape.

II. The Overseer Problem

In short, the OP is the inability of artificial systems to perform independently, that is, without the prior assistance of an intelligent agent to set the parameters of the system's task and to determine when that task is to be considered correct or complete." The task-determination part is somewhat trivial. Humans, after all, usually design artifacts for the purpose of carrying out specific tasks. Task-completion and validation, knowing when the job is done correctly, is another matter.
A. The Overseer and PDP. How does PDP come to know a right or wrong answer? What is the "right" answer to a problem for a system that has just been activated, initially lacking any data to work with? Let's begin with a theoretical neural net in action.

Consider a net designed to distinguish kinds of cars by color and manufacturer. First, the net must be "trained up" from examples of cars. To do this, the system is attached to a video camera in a parking lot. For training, the camera first must be pointed at a car and then the net makes a random guess of color and manufacturer. Actually, at this early stage the net does not have an internal representation of 'color' or 'manufacturer'; the guess is a true shot-in-the-dark. The net's trainer inputs a new signal that either reinforces the current internal representation, in the case of a correct guess, or alters the representation, in the case of a wrong guess. After a large number of examples and correction signals, the net is trained. That is, the system has developed a paradigm model for each color and each manufacturer, respectively. Now the camera can point to any car and the net will determine the make and color.

The key to the OP is the role of the trainer. Of course, the trainer must determine the nature of the problem to be solved. More importantly, the trainer already has knowledge of what constitutes a right or wrong response from the neural net. From the net's point of view, as it were,
one answer is just as good as the next. The net has no
objective reference for determining correct results except by
the feedback of the trainer.

For some readers, this analysis may be a matter of
highlighting the obvious. If so, recall the claims of
strong-AI, such as the opening quote of the introduction:
"We are about to conceive of the knower as a computing
machine." The thrust of the OP is that "the knower," if he
is to be given equal cognitive status to human agents, must
be able to function without an overseer to specify tasks and
predetermine the nature of correct responses. Although I
have found little attention given to this problem, here are a
couple of notable quotes:

Our license to speak of these systems as judging similarity
depends upon the fact they classify together patterns that we
also take to be similar (emphasis mine).\(^5\)

The problem here is that the designer has determined ... that
certain possible generalizations will never be found. All this
is well and good for toy problems ..., but in real-world
situations a large part of human intelligence consists in
generalizing in ways that are appropriate to a context. If the
designer restricts the net to a predefined class of
appropriate responses, the net will be exhibiting the
intelligence built into it by the designer for the context but
will not have the common sense that would enable it to adapt
to other contexts, as a truly human intelligence would.\(^6\)

B. The Overseer and GOFAI. The OP looms larger for
GOFAI. In PDP, the system only needs to be fed "right"
examples. Likewise in programmed digital computers, correct
data must be input (recall the programmer's cliché "garbage
in, garbage out"). Furthermore, the entire structure of the
computer's task must be defined in detail in the body of the program. Right and wrong are, in a manner of speaking, in the eyes of the beholder—in this case the person writing the software code. The OP here is quite stark. The computer does nothing without an overseer-programmer to define a) the problem, b) the nature of correct answers, and c) a detailed algorithm for how the problem is to be addressed. The last requirement, (c), makes the OP stronger for GOFAI than for PDP since, once again, the latter has no program.

C. Objection: PDP Can Overcome OP. The unique abilities of PDP might offer a way to overcome the OP. There is, the objection runs, no need for a program and therefore no need for a programmer-overseer. Inherent in the PDP architecture is a

very simple mechanism for extracting regularities from an ensemble of inputs without the aid of sophisticated generalization or rule-formulating mechanisms that oversee the performance of the processing system. These learning rules are completely local, in the sense that they change the connection between one unit and another on the basis of information that is locally available to the connection rather than on the basis of global information about overall performance. The model thus stands as an alternative to the view that learning in cognitive systems involves the explicit formulations of rules and abstractions under the guidance of some explicit overseer (emphasis mine).7

Unlike a programmed computer, neural nets form conceptual representations apart from any guidance on how this formation is to be done. For example, our own conceptual schemes intended to represent the physical world rely heavily on our five senses. The distinguishing features
of objects are often given in terms of shape, color, size, texture, etc.

In contrast, consider another theoretical neural net that distinguishes trees from telephone poles. This net's input consists of a TV camera, audio microphone, radar, and infrared detector. With this array of choices from which the net will build its paradigm-tree and paradigm-telephone-pole, we have no idea nor control over what "sensory" input the net will decide is significant. The net's determination of significance and its internal representation are completely unknown to any programmer-overseer. In Kantian terms, we do not know what categories the net will develop to distinguish the two kinds of objects.

Anyone who fully understands the OP will see that this attempt to escape the problem will not do. First, an overseer is required to define a problem for the net to solve. Second, even without an algorithm for solving the tree-pole problem, the neural net still requires an overseer to say whether the output during training is right or wrong. Without this feedback, the net cannot build its paradigms. Third, the net's ability to make generalizations can only be exercised if it is given the correct exemplars on which to base its model. Three trees may be sufficient to give the net some prototype of 'tree,' but what if the trainer mistakenly inputs a bush? Then the paradigm is distorted and the net's performance is diminished. In this way the
accuracy of the net's future performance is wholly dependent on being properly trained, where "proper" is once again in the eyes of the overseer.

III. Induction

At this point, I would like to demonstrate the OP in action. The problem of induction has stubbornly refused to succumb since the time of Hume and, with Goodman's help, it has actually intensified. Let's see how the problem of induction (PI) and the overseer problem jointly pose a conceptual snare for strong-AI.

We want computers to make the "right" inferences based on experience. Then again, we hope to make similar right inferences. When are we justified in going from "x β's have all been found to be y" to "all β's are y" or even "it is probable that the next β I observe will be y"? How many observations are required? Since the PI is still unsolved, we can only observe that we do in fact make judgments about the future based on past experience.

How does a computer solve the PI? Answer: An overseer works out the solution in advance. To illustrate, recall from Chapter 2 the "script" technique in programming. Stereotypical situations (e.g., the restaurant script, the ball game script) are given to the computer as guidelines to operate within. However, to describe a given situation as "stereotypical" implies that we already know what regularities are to be expected in the future in such a
scenario. This simply ignores the PI, as we almost always do in our day-to-day routines. A scripted program is a safety net provided by the overseer to insure that the system makes the "right" inferences. Thus the system never faces real induction.

Although things are better for PDP, the overseer is often presupposed in induction problems. For instance, from a given number of exemplars the net will establish that "all Trans Ams have been Pontiacs" and thus "the next Trans Am examined will be a Pontiac." However, if the goal of the training is to get the net to form this "belief," obviously the overseer must have been satisfied in advance that this belief is true.

I should point out that part of the power of PDP is its ability to make inferences and detect patterns not found by humans looking at the same data. For example, banks and insurance companies currently use neural net simulations to detect financial patterns that will help determine loan applications, etc. In some respects, neural nets are better at induction than we are. The reason this ability is insufficient to solve the OP lies in the training. For any net to make inferences, an overseer is still required to train the net with examples that the overseer already knows to be "correct" inductive inferences.

To conclude the discussion on the PI, I would like to address a conceptual issue arising from differences in
education. One reason AIers do not wrestle with induction the way philosophers do is because engineers, mathematicians, and computer scientists approach the subject through probability theory. An informal yet philosophically significant part of probability theory is what I will call the all-things-being-equal wand.

To illustrate, I recall my introduction to probability theory in a communications class. The professor would always talk about the probability of a given event "all-things-being-equal." An example would be, "The probability of this coin coming up heads when flipped is 0.5, all-things-being-equal." We understood that this qualifier meant we could ignore the probability of a bird diving through the window and snatching the coin or the probability of the coin spontaneously decomposing. For all such instances, the professor would always wave the all-things-being-equal wand to eliminate unwanted factors.

Returning to AI, without an overseer providing the all-things-being-equal wand for the computer, the system has no way of "just knowing" what factors can legitimately be ruled out. Somehow all of the students in my communications class understood what factors were eliminated by "all-things-being-equal." Unless computers can develop a similar ability, the overseer will not fade into irrelevance. And unless the overseer fades into irrelevance, strong-AI claims will continue to be exaggerated.
IV. Thesis Conclusion

There is a danger in criticizing AI that I have tried to avoid in this work. Too often critics point to what computers currently fail to do without a view toward advancing technology. This tactic is a trap: Picking at the difficult technological barriers and hardware-software limitations faced by AI today will inevitably backfire.

For example, digital computers have had a longstanding difficulty in processing and recognizing images. Given sufficient computing time, most systems could be programmed to recognize faces; however, we have the ability to recognize a given face almost immediately. For years critics like Hubert Dreyfus have cited "elementary" perceptual mistakes made by computers that the average child could avoid. Today, computer image processing is quite advanced. Consider the accuracy of Tomahawk cruise missiles used in the Gulf War. With the help of a maturing technology in charge-coupled devices, many of Dreyfus' early criticisms about computer perception have now been met.

Have I avoided this trap? I'm not sure. Unless digital computer technology gives way to PDP, the Chinese Room will retain its force (see note 3). The Frame Problem may very well be solved or at least made much less noticeable in time and I do not claim that it is an insurmountable difficulty. The Overseer Problem, however, is highly conceptual and will not likely fall in the wake of new
technology. If I have left myself vulnerable to the trap, so be it.

What has been accomplished in this thesis? Instead of simply recapping my arguments from Chapters 1-3, I would like to address strong-AI in general. The three chapters individually raise questions that must be answered before AIers can claim that computers are intelligent. I do not mean that AI research should come to a stop until computer scientists can appease every philosophical nemesis. However, I do think these concerns must be addressed before AIers continue to lob conceptual bombs into metaphysics, the philosophy of language, and particularly the philosophy of mind.

What has not been accomplished in this thesis? I have not developed criteria for intelligence that clearly demarcate man from machine. Such criteria would draw the discussion far afield and would need to consider the problem of other minds and perhaps Wittgenstein's private language argument, especially where the overseer/trainer is involved. My goal was the lesser task of tightening the reins on some in the AI community. AI has unquestionably helped in the development of the necessary conditions for intelligence. Strong-AI claims, however, lead the reader to believe that computer science has already determined the sufficient conditions for intelligence. This claim, I have shown, is too optimistic.
Finally, I urge readers from all disciplines to keep track of their metaphors. Every advanced field of study incorporates rather innocent sounding words into the jargon of the field. In the case of computer science, words like 'sees', 'knows', and 'memory' are used to describe computers because we know what such words mean when applied to people. We use these words metaphorically, which is perfectly fine. Too often the metaphor is lost. Common words with technical meanings start migrating between fields and then back to ordinary speech. It is regrettable when the man-in-the-street becomes confused by this migration. When the scholar and the scientist likewise become confused, entire worldviews may hang in the balance.
NOTES


3A prescriptive syntax, like a computer program, drives the system. Executing a program is synonymous with following a set of syntactic rules. Without a program, PDP has no driving, prescriptive syntax; however, its internal mechanism may be described syntactically. Although Searle believes the Chinese Room is applicable to PDP since there is a syntactic description available, I believe the Chinese Room fails unless the syntax is prescriptive.

4John Searle has mentioned a similar problem to OP called the "Homunculus Fallacy," but so far he has applied it only to digital computers. See Searle, "Is the Brain a Digital Computer?" APA Proceedings 64 (November 1990): 21-37, especially pp. 28-29.

5Bechtel, 27.


SELECTED BIBLIOGRAPHY

Books


Articles


____________. "Minds and Brains without Programs." In Mindwaves: Thoughts on Intelligence, Identity, and Consciousness.
Koperski  82


