

Philosophy Concerning Artificial Intelligence

by Jennifer C. Lorber
Henry Rutgers Scholar 1984-85

A thesis submitted to
Rutgers College
in partial fulfillment of the requirements
for the degree of
Bachelor of Arts

Written under the direction of
Professor Allen Ginsberg
of the department of computer science

New Brunswick, New Jersey
May, 1985

TABLE OF CONTENTS

1. INTRODUCTION TO ARTIFICIAL INTELLIGENCE	3
1.1. AI Applications	3
1.2. State Space Representations	6
1.3. Can Computers Be Intelligent?	8
2. KNOWLEDGE	11
2.1. The storage of information	11
2.2. Knowledge Representation	13
2.3. Epistemology	15
2.4. Conceptual Structures	20
2.5. Perceptual Mechanisms	21
3. EXPERT SYSTEMS	23
3.1. Knowledge Engineering	23
3.2. Rule Formation	25
3.3. Control Strategies	29
4. INTELLIGENCE	35
4.1. The Mind-Body Problem	35
4.2. Behaviorism	36
4.3. The Identity Theory	38
4.4. Functionalism	39
5. CONCLUSION	43
5.1. Objections	43
5.2. A Theoretical Obstacle	45
5.3. Final Comments	46

LIST OF FIGURES

Figure 1-1:	<i>A State Space Representation of a Computable Problem</i>	7
Figure 1-2:	An example of SHRDLU's blocks world	9
Figure 2-1:	Frame Representation of a Classroom Situation	13
Figure 2-2:	Frame Representation of a Teacher and Blackboard Constructs	14
Figure 2-3:	The Flow of Information	18
Figure 2-4:	Mechanisms of Perception	22
Figure 3-1:	Control Strategies	31

LIST OF TABLES

Table 3-1:	<i>Simplified Representation of a Precedence Rule</i>	26
Table 3-2:	<i>Example of patient data represented in precedence rule chart</i>	26

PREFACE

The comparison of Philosophy and Artificial Intelligence (AI) has often effected discovery of theoretical parallels. As John McCarthy observed:

Philosophy has a more direct relation to Artificial Intelligence than it has to other sciences. Both subjects require the formalization of common sense knowledge and repair of its deficiencies. [28]

The following thesis attempts to strengthen such parallels. The connections between contemporary Philosophy of Mind and research in Artificial Intelligence are not restricted to knowledge representations such as predicate logic. Indeed, there exist strong similarities between AI's *problem-solving formalisms* and Philosophy's *conceptual structures*. Genuine understanding of these similarities is contingent upon genuine understanding of both types of theoretical entity. This paper provides an introduction to research in both Philosophy and Artificial Intelligence. It concentrates on those aspects of each domain which contribute to an understanding of their theoretical parallels.

- Chapter One is an introduction to Artificial Intelligence.
- Chapter Two is an introduction to knowledge representation schemes.
- Chapter Three is an indepth analysis of the most successful type of AI programs: expert systems.
- Chapter four is an attempt to formalize the philosophical attribution of intelligence.
- Chapter five presents the conclusions of this thesis.

CHAPTER 1

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

1.1. AI Applications

The following scenario depicts the current state of technology in Artificial Intelligence:

Carol was running late for work. Of course this was the morning her company car did not start. After several unsuccessful attempts, she called Tracy, the company mechanic.

"What did FAE say?" asked Tracy.

"FAE?"

"Ford Automobile Expert, it's on the right side of the dash." Tracy seemed impatient. "Go run the program and call back if it's something you can't fix." Tracy hung up.

Carol stared at the phone for a minute. Then she went back to the car. She was mildly surprised to find that the flip side of the glove box was a keypad. Inside was FAE, who asked

WHAT TYPE OF CAR PROBLEM DO YOU HAVE?

On the screen was a list.

1. CAR WON'T START
2. FLAT TIRE
3. OTHER CAR PROBLEM

Carol pressed '1' on the keyboard.

SIMPLE CHECKS:

1. HEADLIGHTS ARE DIM
2. FUEL FILTER CLOGGED
3. BATTERY CABLES LOOSE/CORRODED
4. NONE OF THE ABOVE

This confused Carol. Not knowing what else to do, she typed,
 "Where is the fuel filter?" FAE responded with a diagram of the engine.
 FOLLOW THE HOSE FROM THE CARBURETOR TO A BROWN DISK.
 SEE IF ANY GAS CAN GET THROUGH.

"What do I look like, a mechanic?" But Carol easily completed the
 simple checks and answered 'none of the above'.

STARTER DATA:

1. NO CRANKING
2. SLOW CRANKING
3. NORMAL CRANKING
4. GRINDING NOISE FROM STARTER

Carol hoped all the rest of the questions were this easy. She typed
 '3'.

GAS GAUGE READS EMPTY?

"I am not stupid." Carol typed. She wondered what she why she felt
 insulted by a car.

I ASSUME THAT MEANS NO. TYPE 'FIX' TO CHANGE YOUR
 ANSWER.¹

ODOR OF GASOLINE IN CAR:

¹ Actually a computer can not yet understand what Carol meant by her remark. However, it can be programmed to recognize negatives.

1. NONE
2. NORMAL
3. VERY STRONG

The car did seem to smell like gas and Carol typed '3'.

OUTDOOR TEMPERATURE IN DEGREES FARENHEIGHT:

It was getting hot in that car. Carol guessed it must be at least 75.

INTERPRETIVE ANALYSIS

DIAGNOSTIC STATUS :

CAR FLOODED

TREATMENT RECOMMENDATIONS:

WAIT TEN MINUTES OR DEPRESS ACCELERATOR
TO THE FLOOR WHILE STARTING

Ten minutes later, Carol was on her way to work.

The general model for an automobile mechanic program was developed at Rutgers merely to illustrate to students the use of EXPERT, a tool for the design of expert systems. Extensively used expert systems include: PUFF, which performs pulmonary function tests; R1, which determines correct VAX computer configurations; DENDRAL, which infers the molecular structure of unknown chemical compounds; MACSYMA, which performs symbolic integration; and a Serum Protein Diagnostic program which interprets serum protein electrophoresis tests. [47, p.2] Expert systems are the most successful AI programs developed so far, but current research in AI includes speech understanding programs, visual information processors and robotics. All AI programs are similar in that success depends on a good organizational structure of the knowledge base. [35, p. 365]

1.2. State Space Representations

In storing information, program designers must make a distinction between the representation of facts (true rules and propositions about the problem domain) and the use of rules for manipulating information to produce a solution. This distinction is reflected in the customary characterization of programmable problems as state space representations. A problem state is designated as four sets:

1. A set of all possible configurations of all relevant variable components of the problem domain. Call it Σ .
2. A set of rules of the form: IF A THEN B, where A is a member of Σ , and B is a different member of Σ .
3. A subset of Σ containing those states from which it possible to initiate problem solving activity. These are called *initial states*.
4. A subset of Σ containing those states which define a solution to the problem. These are called *goal states*.

Problem solving may be viewed as a search which starts at one of the initial states. Rules whose left-hand-side match the initial state are used to generate successor states. A *heuristic evaluation function* may be used to evaluate each successor state and indicate which are most likely to lead to a problem solution. Those states with the highest probability of generating a goal state are expanded to produce a new set of successors. The process is repeated until either a goal is generated or the set of successor states do not match the left-hand-side of any rules. In the state space representation of figure 1-1, there are three rules which have the designated initial state on the left-hand-side. Of the three successor states, two are evaluated by the heuristic function as being close to a solution. These are then expanded according to applicable rules and the heuristic function is used to evaluate their successors.

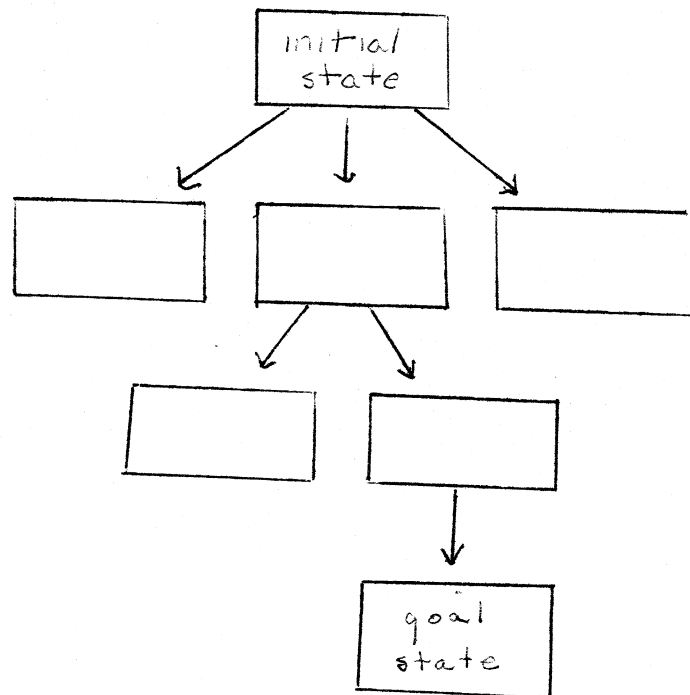


Figure 1-1: *A State Space Representation of a Computable Problem*

This process ends at the third generation when a goal state is generated. The reasoning process which led to the solution is documented as the series of rules which lead to the goal state. When the current set of successors cannot be expanded with the given set of rules, the search fails and the problem remains unsolved. Much of the pioneer work in Artificial Intelligence was concentrated on search methods. Recently, however, attention has turned to the organization of state spaces in order to maximize the efficiency of the search. [30, p. 65]

1.3. Can Computers Be Intelligent?

Because the majority of problems which humans face daily cannot at present be stated so formally, one may intuitively conclude that computers cannot be said to *know* or to be *intelligent*. Philosophers as well as computer scientists are prone to the use of *hypothetical* examples designed to capture the intuition and extend its aura of certainty to every instance of machine problem solving. However, I find that actual research in computer science easily affords a basis for discussion of the possibility of machine intelligence. One example could be SHRDLU. SHRDLU is a program which simulates the conversation of a hypothetical robot. This robot is designed to manipulate a finite set of boxes. SHRDLU internally represents a formation of boxes (see figure 1-2). It can answer questions about the relations between them (i.e., "Is the small green box under the big red box?"). It can change its internal representation as if it had moved some boxes (i.e. "Put the red box on top of the blue box." "OK"). SHRDLU has some very obvious limitations. [5, p. 122-23] It cannot accept ungrammatical input. It produces very rigid responses. It is not capable of hypothetical or analogical reasoning. However, if one takes information to be the mental representation of justifiable propositions, then SHRDLU can truly be said to possess information about its limited domain. SHRDLU's reasoning processes are linguistic and deductive; it can derive information concerning the relations between the blocks. This ability to take in information, reason with it and produce information based on it indicates that SHRDLU *understands* the block world. [3, p. 22] It is through discussion of such research and not through discussion of hypothetical programs that we discover the problems involved in constructing computer intelligence.

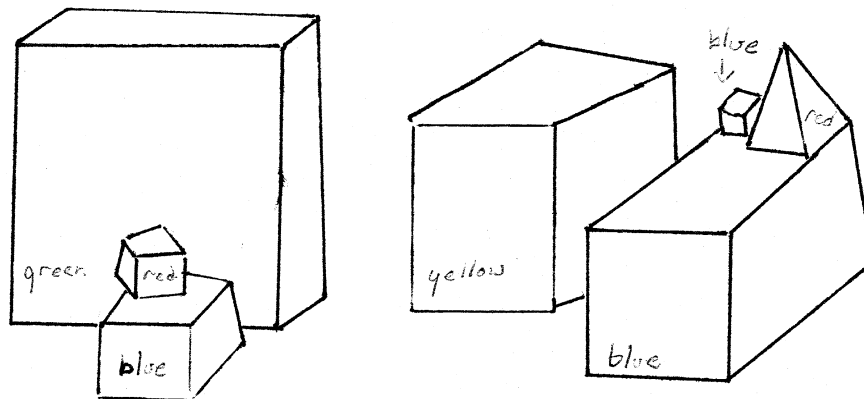


Figure 1-2: An example of SHRDLU's blocks world

Though SHRDLU makes a strong case for the existence of computer understanding, one may still maintain that computers will never be intelligent because they lack some elusive 'analogue' attributes or because they understand only very limited domains. [13] It becomes necessary to look beyond the point which SHRDLU proves about computer understanding to questions about how humans understand the external world. Davidson maintains that one cannot attribute human qualities to a machine unless that machine shares our perspective on the information we have in common. [8, p. 345] A strong supporter of research in Artificial Intelligence may maintain not only that an appropriate program will enable a computer to *understand* things, but also that the program will serve as a *explanation* of human understanding. [37, p. 282-83] But what sort of explanation *is* a computer program? One way to view a program as an explanation of understanding (perhaps you can think of others) is to adopt the view that the ability to consistently perform operations on a variety of symbols constitutes

understanding. This argument is supported by the fact that systems which lack this ability are unable to identify or affect symbols in a proscribed manner; and this intuitively constitutes confusion. However, conclusions concerning computer intelligence must be subject to standards stricter than intuition.

CHAPTER 2

KNOWLEDGE

2.1. The storage of information

One safe assumption concerning knowledge is that it has something to do with facts, or with information conceived as true representations of facts. To communicate knowledge is simply to communicate information. But this does not immediately imply the identity of the two concepts. It merely opens an avenue of study through which one may proceed to the goal of characterizing what it is to know. A computer represents information in the form of *bits*. Bits are instantiated by electronic circuits whose voltage levels are ascribe one of two values, zero or one. Combinations of bits may be conceived as strings of zeros and ones; they are binary representations of machine instructions and data. Instructions of the type:

```
00010000001110100000000011111111
```

are typical of *machines languages*. They may be entered directly into the memory of a computer and will immediately be executed by the machine's *hardware*. Hardware denotes integrated circuits, printed circuit boards and the rest of the physical components of the machine. It is so called in opposition to *software*. Software denotes abstract problem-solving strategies or programs which are representable in a variety of forms, such as punched cards, tapes and discs. It consists of sequences of high level instructions, such as:

```
IF (statement1 = true) THEN (statement2 = true)
```

However, all software must be either translated or interpreted into the machine's binary code before it can be executed by the machine's hardware. It is theoretically possible to implement such instructions directly in the hardware, but their variety renders such an attempt both costly and inefficient. Nevertheless, the distinction between a machine's hardware and its software is simply a matter of translation or interpretation. The software directly represents causal interrelations in the machine's hardware. It is often referred to as a *virtual machine* whose machine language is the set of high level instructions of which the program is composed. Tannenbaum observes that a machine's hardware and its software are *logically equivalent*. [44, p. 11] They represent two different perspectives through which to view the machine, reminiscent of Spinoza's distinction between thought and extension.

The fuzzy distinction between hardware and software presents an obvious analogy for the human brain. Stephen Kosslyn argues that the construction of a computer has influenced intuitions on theories of self, and philosophy of mind has suffered in the process. [23] It is true that the general term *information-processing-system* has blurred the distinction between information stored in a computer and information stored in the human brain. For many, the distinction is in the circuitry which acquires and synthesizes the information. As Dreyfus observes, "Programmers rush in where philosophers...fear to tread." [13, p. 183] Ever more sophisticated technological breakthroughs in computer hardware have pragmatically optimistic computer scientists analyzing human problem solving activity in the hope of mapping those procedures into computer programs.

2.2. Knowledge Representation

One universally agreed upon fact among researchers in artificial intelligence is that intelligent problem-solving activity requires large amounts of knowledge about the problem domain. [35, p.5] Thus, the level of intelligence attributable to a computer will always be constrained by the amount of problem relevant information representable by the system. This is referred to in computer science literature as *The Knowledge Representation Problem*. Attempts to deal with this problem are many and varied. I will describe two: frames and scripts.

A frame is an information-storing structure which represents answers to common questions about a hypothetical stereotypical situation. [29, p. 109] For example, a frame designed to represent a classroom situation may look like figure 2-1. The slots may be filled with frames which represent available information about a certain construct or they may be filled with default values when no information is available. For instance, the teacher and blackboard slots may be filled with the frames in figure 2-2, and these may be filled with other frames or with default values when the data is unavailable.

TEACHER	DESKS	BLACKBOARD
STUDENTS	NOTES	AUDI-VISUALS

Figure 2-1: Frame Representation of a Classroom Situation

A control strategy designed to fill the empty slots may search for or request information necessary to fill a certain frame.

PERSON	NUMBER	SIDE-OF-ROOM?
KNOWLEDGE-BASE	DUSTY?	FRONT-OF-ROOM?
ANIMATED?	CHALK	BACK-OF-ROOM?

Figure 2-2: Frame Representation of a Teacher and Blackboard Constructs

A script may be conceived as a highly specialized frame. It represents patterns of causal relationships which occur in a specified sequence of events. Scripts serve to provide assumptions concerning such sequences, to provide information concerning the relations between events and to focus attention on unusual events. The paradigm example is of a script which represents a visit to a restaurant. [35, p.234-37] Given only the statement:

S went to a restaurant, had a hamburger and left.

a restaurant script would enable a computer program to answer a wide variety of questions including:

- Did S sit down?
- Did S eat?
- Did S talk to the waitress?
- Did S pay his check?

Frames and scripts have both been labeled as *declarative* representations of knowledge. This reference is in contrast to a *procedural* knowledge representation, which entails fewer straightforward facts and more procedures to manipulate them. SHRDLU, for example, represents the blocks world with the same procedures it uses to manipulate the blocks. [35, p.240] However, no system can employ either declarative or procedural representation to the exclusion of the other, and there is no clearcut boundary between the two types of representation.

2.3. Epistemology

Information storage and processing methods are of much concern to modern epistemology. Traditional attempts at defining knowledge have concentrated on identifying necessary and sufficient conditions for what it is to know. In general, these conditions have been of the form: [17, p.35]

S knows that P if and only if

1. P is true
2. S believes that P
3. S is justified in believing that P

Traditionally, *justified true belief* has meant "having the right to be sure", or something very similar. [2] Edmund Gettier argues that justified true belief is not an adequate characterization of knowledge because he is able to formulate counterexamples which refute any such claim. For instance: [17, p.36-37]

Suppose that Smith and Jones have applied for a certain job. And suppose that Smith has strong evidence for the following conjunctive proposition:

- (d) Jones is the man who will get the job, and Jones has ten coins in his pocket.

Smith's evidence for (d) might be that the president of the company assured him that Jones in the end would be selected, and that he, Smith, had counted the coins in Jones' pocket ten minutes ago. Proposition (d) entails:

- (e) The man who will get the job has ten coins in his pocket.

Let us suppose that Smith sees the entailment from (d) to (e), and accepts (e) on the grounds of (d), for which he has strong evidence. In this case, Smith is clearly justified in believing that (e) is true.

But imagine, further, that unknown to Smith, he himself, not Jones, will get the job. And, also, unknown to Smith, he himself has ten coins in his pocket.

The conclusion is that Smith is justified in believing that statement (e) is true, but he does not *know* it.

Though Gettier's arguments are valid for the traditional analysis of justification, justified true belief is not an inherently inadequate characterization of knowledge. The issue which prevents its acceptance is one of standards for justification. One justification which avoids Gettier's dilemma is proposed by Fred I. Dretske. His characterization of knowledge is based on empirical facts concerning the communication of information.

To understand Dretske's theory, one must be able to view information in quantitative form. The amount of information generated at a source is measured as differentiations are made between two competing alternatives. In Dretske's example, one of eight employees has been nominated for a special task. The information whose content specifies the nominee (assuming each employee has an equal probability of being chosen) requires three differentiations:

1. The nominee belongs to one of two groups of four.
2. Of that group, he/she belongs to one of two sets of two employees.
3. The nominee is one of the two in that set.

The amount of information involved in specifying the nominee is thus three *bits*. Each bit will be a one or a zero, to signify the result of one differentiation. In general, the amount of information at a source is the logarithm in base two of the number of competing alternatives. [12, p.5] Were there 16 employees to choose from, the amount of information embodied in the specification of the nominee would have been 4 bits. In Dretske's account of communication theory, a source of information usually carries some degree of equivocation. The larger the number of competing alternatives, the larger the equivocation, the larger the value of information generated. [12, p. 62] The amount of information generated, however,

is not necessarily proportional to the amount of information transmitted. Were I to inform you which of the eight employees were chosen, I would relay the amount of information generated minus the *equivocation* at the source. There would no longer be any alternatives to consider. However, I must communicate that information by some method. I may write the information on a piece of paper, send you a message at your computer terminal, phone you, or even relay it by Morse code. In any case, your receipt of that information may be affected by the communication medium. Dretske calls information you receive that is not generated by the source *noise*. Possible scenarios in the communication of information are depicted in figure 2-3.

This account of information need not be as exacting as its presentation implies. Dretske does not attempt to define information, merely to characterize it in terms of its alternatives. To calculate an amount of information, one must list the alternative possibilities, the probabilities associated with each alternative and the conditional probabilities among the alternatives. For most situations, this is not possible. Yet neither is it necessary. The amount of information generated at a source depends on the related probabilities which *exist*, not upon those which we can independently verify. [12, p.56] Information is factual. Its receipt depends on its content relative to the receiver's prior evaluation of alternatives. For instance, if you know that none of the four employees in the first group chosen were chosen, then you require just two bits of information to identify the nominee; whereas someone else may need three bits to receive the same information.

By now, it may be clear how Dretske is able to use this characterization of information to characterize knowledge. Information by definition carries truth.

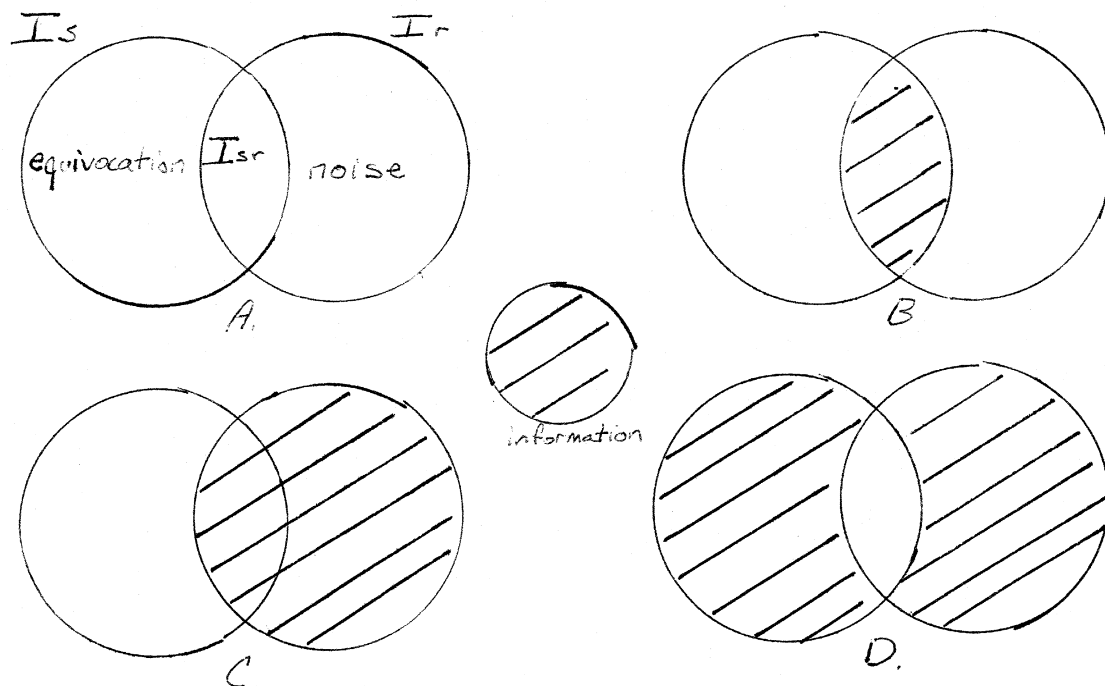


Figure 2-3: The Flow of Information

- A. I_s = information at source
 I_r = information recieved
 I_{sr} = amount of information recieved
from information at source
 - B. perfect communication of information
 - C. partial communication of information
 - D. lack of communication of information
- [12, p.16-18]

Therefore, a person may *know* that my name is Jennifer if his/her belief that my name is Jennifer is *causally sustained* by the *information* that my name is Jennifer. [12, p. 86] An example clearly illustrates Dretske's thesis: Martin may be caused to believe that the door is open without receiving the information that the door is open, but upon receipt of the information that the door is open (e.g., looking at the open door), his belief is sustained, and he *knows* the door is open.

Though this analysis of knowledge clearly answers Gettier, the skeptical response to it cannot be brushed aside. A skeptic will focus on the channel of communication between source and receiver and conclude that it cannot be assumed to function properly enough to guarantee the accurate transmission of information. I do not here refer to the noise which results from the communication of information. For information of this type is redundant from the point of view of the receiver. The channel to which the skeptic objects is a telephone wire or a person's vision: a channel must function as a fixed framework which admits of no relevant alternative states. [12, p.123] If relevant alternative states are known to exist, we must eliminate them through proper testing and monitoring. To say that all communication channels are inadequate is to say that all communication is impossible. It remains with the skeptic to prove that an accepted channel is creating misinformation. [12, p.131]

Still a third argument may state that a receiver is not capable of processing all the information that is transmitted. This assertion is accepted by Dretske, provided the specified signal is transmitted in analogue form. A signal in analogue form will always contain more determinate information about its source than is possible to process. For to process information, one must extract the information from the signal and identify the source. This digitalization process allows us to recognize two sources as being *essentially* similar. [12, p.141] That is, to accomplish the attribution of identity to external objects. The analogue-to-digital conversion typical of information processing activity constitutes cognition, or knowledge acquisition. The conversion may be from a visual image to a familiar face or from a pressure gauge to a numerical measurement. There is always more

information available at the source than it is possible for a system to abstract and categorize. [12, p.153]

2.4. Conceptual Structures

A source of information may be any object of perception. What qualifies as the object of a perception is the most specific piece of information generated at the source. Informational relationships between an object of experience and its causal antecedents or constant corollaries are determined by the cognitive capacities of the information processor. For instance, we perceive a doorbell ringing, but we do not perceive a finger on the button. The relationship between a finger on the button and a doorbell ringing is embodied in conceptual structures which have been formed through cognition. We may thus *learn* about interrelationships between sources of information, but a causally sustained belief that the bell is ringing does not qualify as knowledge that someone is at the door unless the relevant semantic structure admits of this relationship. The content of a belief is a mere digitalized representation of information, while a conceptual structure may be viewed as a cluster of interrelated beliefs.

Once constructed, a conceptual structure is used to process information. Its *meaning* derives from the type of information which led to its development. Its function is to classify perceptual tokens of the same type. Dretske claims that all learning progresses in this manner. In attempting to teach a concept, we endeavor to accurately transmit the information embodied in it in hope that the pupil will form a similar conceptual structure. [12, p.195] Once the structure has been formed, its causal origin is lost, for subsequent tokens may arrive in a variety of different situations. This flexibility inherent in conceptual structures may allow for some

inaccurate classifications. Tokens may be identified as a type exclusive to a particular conceptual structure despite its lack of genuine information to that effect. False beliefs are the result of such false categorization. [12, p.193]

2.5. Perceptual Mechanisms

Dretske's theory on conceptual structures has strong support from theoretical psychology. Contemporary research has demonstrated that the brain temporarily stores sensory input in the form of an icon. Because these last for such a short time (250 milliseconds), brief glances would produce no visual image if a person did not have preconstructed models of innumerable visual images. Such mental models are called *schema*. [42, p.27] Schema are formed from mental models of smaller dimensions called *percepts*. Psychologists have identified various perceptual mechanisms in the brain whose function is to store and retrieve information in the form of schema (see figure 2-4). The *associative comparator* retrieves stored percepts which best match all or part of a perception, or icon. It has the ability to attribute identity to external objects. [42, p.31] Another type of perceptual mechanism is an *assembler*. It combines the recalled percepts into a schema. Familiar images will not require the use of primitive percepts, since a corresponding schema may already exist in the brain. Unfamiliar images will require repeated eye movements or other motor mechanisms working in cooperation with the assembler to establish the correctness of the completed model. [42, p.32] Correctness of the model will vary with the amount of attention paid to specific areas of the object of perception.

Analysis of complex behavior, such as chess playing, demonstrates the need for schema which organize fundamental percepts into recallable patterns. [42, p.44]

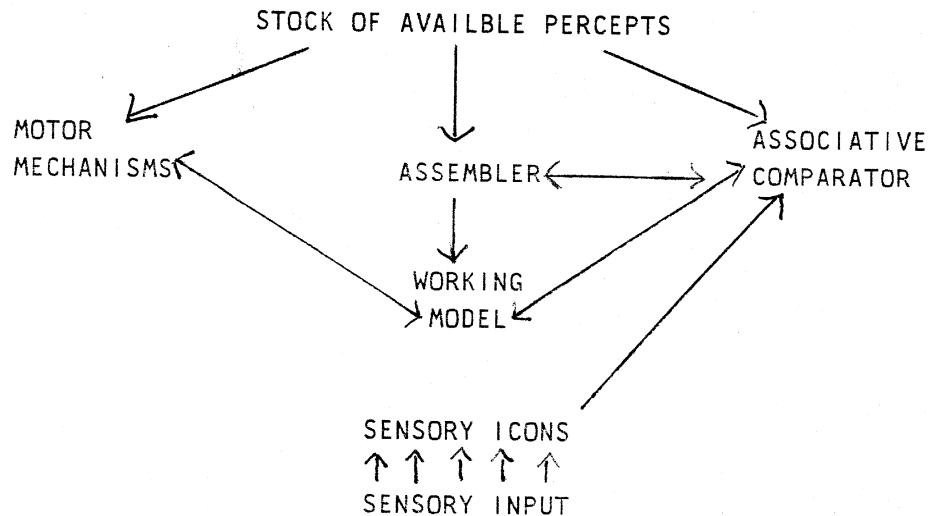


Figure 2-4: Mechanisms of Perception
[42]

Several studies have shown that, while a novice chessplayer must analyze the board to determine the correct move, a chess master need only glance at it. The master recognizes a greater variety of patterns and can assemble patterns more quickly. Furthermore, internally generated images have the same nature as sensory icons and thus may be reinterpreted by the same perceptual mechanism. [42, p.61]

It is interesting from the point of view of this thesis that the assembler and the associative comparator operate totally independently in the brain. The biological disruption of one mechanism has no effect on the performance of the other. [42, p.33] Such is also the case with the storage and retrieval procedures in a computer program.

CHAPTER 3

EXPERT SYSTEMS

3.1. Knowledge Engineering

Many projects now classified as expert systems began as basic research in knowledge representation. They evolved into realistic schemes for problem solving. [47, p.157] No one will disagree that a person whom we call an expert is one to whom we would attribute intelligence. An expert is a person whose extensive experience and specialized knowledge renders him a source of information which no textbook could ever capture. The phenomenon of human expertise amounts to procedures for applying knowledge to practical situations. A distinctive attribute of expert reasoning is the ability to use available facts about a problem to narrow the scope of possible solutions. [47, p.3] It is rarely easy for an expert to document personal methods of solving problems in the domain of specialization. But with the help of an expert system designer, or *knowledge engineer*, it is possible for an expert to formulate well-defined rules and programmable techniques which allow computers to serve as intelligent assistants and advisors in fields where human experts are scarce.

An expert system is a computer program which meets the following general specifications: [47, p.6]

1. The problem solutions or advice are enumerable.

2. There exists a reliable method of collecting data about the problem.
3. It is assumed that we deal with a limited domain.
4. There exists logical or probabilistic rule-based knowledge which is capable of linking data to conclusions.
5. There exists a reasoning control strategy capable of accepting information from the user, asking questions of the user to supplement the knowledge base, and explaining reasoning processes when asked.

Though it is an extremely difficult task for the knowledge engineer to acquire the knowledge necessary for encoding rules from the expert, the manner in which to encode those rules presents an even more difficult set of problems. The constraints imposed by computer representation of knowledge involve a tradeoff between the power required to accurately express expert knowledge and the simplicity required to explain, update and describe the knowledge represented. [47, p.12] *Power* here denotes reasoning techniques; they are seen by knowledge engineers as independent of the knowledge required to solve the problem. Reasoning methods may consist of interrelated subproblems, such as the following: [47, p. 18]

1. Given a pattern of evidence, infer possible diagnoses and degree of certainty possible for each.
2. Given the same pattern, deduce what other types of evidence would strengthen a particular diagnosis.
3. Given a tentative diagnosis, account for unexpected or incompatible evidence.
4. Given a tentative diagnosis, form a conclusive one.
5. Given a conclusive diagnosis, choose the appropriate action.
6. Given conclusions and recommendations, explain the reasoning process which led to their choice.

3.2. Rule Formation

Rules derived from experts may be deductive, inductive or simply unspecified associations. They may link evidence to other evidence, evidence to hypothesis, hypotheses to other hypotheses, or even combinations of evidence and hypotheses to further hypotheses. Rules are most often of the form: IF A THEN B. They are called *production rules*. But other types have been developed. John Kastner's *precedence rules* enable a knowledge engineer to present the logical structure of his program in a form which is easily understood and corrected by an expert. [22] A precedence rule may represent, and thus replace, numerous production rules with a matrix which can be presented to the expert in the form of a chart. Kastner's examples are in therapy planning. He listed all possible treatments for a diagnosed disease in order of their efficacy as a cure. Specifically, drugs known to be effective against disease 'X' may be listed as follows:

DRUG A < DRUG B < DRUG C < DRUG D

'<' means '*is more effective than*'. Kastner also listed the contraindications associated with each drug; that is, the reasons why a drug should not be used for a particular patient. These two lists form the columns and rows of a matrix. For each drug, the relevance of a contraindication is represented in the matrix as a boolean value (see Figure 3-1). To determine which therapy is best for a patient, his individual findings are presented in the same matrix format (see Figure 3-2).

A true value in the patient matrix for a certain drug-contraindication combination prompts a reordering of the drug list. The contraindicated drug is moved to the end of the precedence list. In this example, the patient is mildly allergic to DRUG D, but DRUG D is already on the end of the list, so no reordering is required. The next contraindications then are evaluated. Because the patient is

contraindications/DRUGS	most effective <<< least effective			
	DRUG A	DRUG B	DRUG C	DRUG D
pregnancy	F	F	F	T
resistance	T	T	T	T
severe allergy	T	T	T	T
mild allergy	T	T	T	T

Table 3-1: *Simplified Representation of a Precedence Rule*

[22, p.38]

contraindications/DRUGS	most effective <<< least effective			
	DRUG A	DRUG B	DRUG C	DRUG D
pregnancy	F	F	F	F
resistance	F	T	F	F
severe allergy	T	F	T	F
mild allergy	F	F	F	T

Table 3-2: *Example of patient data represented in precedence rule chart*

severely allergic to both DRUGS A and C, they are moved to the end of the list in the order in which they are evaluated. The new list looks like this:

DRUG B < DRUG D < DRUG A < DRUG C

Further evaluation shows that the patient is resistant to DRUG B and a new list is formed:

DRUG D < DRUG A < DRUG C < DRUG B

No more contraindications are indicated, so the precedence rule has determined that

DRUG D is the best available treatment for this case. Graphic problem space representations of this nature allow a knowledge engineer to discuss his system model with the expert in non-technical language. Moreover, precedence rules are more concise than production rules. They are more easily modified as new variables are discovered in the problem space.

Refinement of a system model is usually accomplished by following an expert around for few days or weeks - observing his behavior and asking questions. The knowledge engineer may start with a few simple problem instances and revise his program to account for an increasing variety of cases. An alternative to this case-by-case approach to rule formulation has been developed by G.A. Drastel and C.A. Kulikowski. [11] Their system starts with textbook representations of the problem space and generates rules based upon taxonomic, causal, and strictly relational links between a specific set of findings and their corresponding physical states. Drastel and Kulikowski view rule formation as a pattern recognition process. All data associated with a certain physical state is placed at the left-hand-side of a conditional statement and the identifier for that state is placed on the right. It is determined which physical states necessarily imply the existence of another and this data is combined with the antecedents for the state implied. This system may also search for evidence which will strengthen the validity of a given rule. Drastel and Kulikowski assert that rules derived in this manner are more likely to correctly classify new cases and will be more easily understood by the expert once he is drafted into the model building process. [11, p.1]

A major limitation of these 'generate and test' methods of rule formulation is that people other than the original knowledge engineer find it difficult to understand

the conceptual model upon which a program is based. William Clancy criticizes such programs because a problem solving strategy and a problem classification are often embedded in the same rule. [7] To trace and modify such rules is not an easy task. Clancy asserts that a domain-independent problem-solving strategy is a prerequisite for an easily understood and modified system. One such system is EXPERT (see page 5). EXPERT is a multipurpose framework for designing expert systems. As Clancy asserts, the existence of this type of system has rendered rule modification an easy task. A program called SEEK has been written which compares the conclusions of an expert system written with EXPERT, to a controlled data base of cases and rates the system's performance. [31] The cases are then analyzed in an attempt to improve that performance. Rules utilized in drawing conclusions are listed for each case. When the program's conclusion is not the same as that of an expert, SEEK has two alternatives. It may search for a rule which is close to being satisfied for the case, and which would, if satisfied, enable the correct conclusion to be derived. Then SEEK may recommend to the EXPERT user that the rule be generalized in order to override the currently dominant conclusion. SEEK's other alternative is to suggest that specifications be added to make the currently accepted conclusion less dominant in all cases. An EXPERT user may experiment with changes without affecting the original program. An extension of SEEK. SEEK II, is capable of refining rules based on experience *without* interaction with a user. [20]

Continual modification is a necessary feature of expert systems development because human experts are not consistently conscious of their reasons for making decisions. For example, probability measures reported by experts rarely turn out to be accurate when conditional probabilities and actual problem solutions are taken into

account. Schemes which weight evidence numerically render it difficult to explain and correct erroneous conclusions. Many expert systems designers cite these and other reasons for avoiding significant use of *strict likelihood measures*. [47 ,pp. 28-38] However, the use of *weighted confidence measures* often serves to give a more concise representation of expert knowledge. They range from -1 (= no confidence) to +1 (= absolute confidence). They are sometimes called *belief measures*. The expert system MYCIN, for example, associates two belief measures with every assertion it considers: a measure of belief (MB) and a measure of disbelief (MD). Both MB and MD are represented by numbers between one and zero. Each MB/MD pair generates a certainty factor (CF) for a given assertion where $CF = MB - MD$. Multiple sources of information concerning the probability of one assertion are combined so that some proportion of smaller CFs are added to the largest to produce a certainty factor larger than that of any individual CF in the set. [35, p.194] However, Weiss and Kulikowski perceive dangers inherent in weighted mathematical scoring. The only belief measure which they will attribute to an assertion is only as large as the smallest weighted evidence or hypothesis that led to its consideration. [47, p. 96]

3.3. Control Strategies

Once rules and other relevant information have been identified, their implementation requires the design of a control strategy suited to the problem at hand. Our paradigm problem solver, heuristic search (see page 6), was an example of a *data driven* control strategy.² Rules are activated if and when their left-hand-

² Synonyms for data driven control are bottom-up, forward chaining, pattern-directed, or antecedent reasoning.

side conditions are satisfied. A data driven control strategy may apply rules as they become activated, or maintain an ordered list of activated rules so as to apply them in the most efficient manner possible. A second type of control strategy is *goal driven*.³ This strategy is to examine all rules whose right-hand-side correspond to the desired goal until it compiles a set of true statements which may represent an initial state. Like the data driven control strategy, a goal driven control strategy may apply rules as they become activated, or maintain an ordered list of activated rules so as to apply them in the most efficient manner possible.

Both data driven control and goal driven control are advantageous for some types of problem but disadvantageous for others. In a situation where there are many possible findings, but few possible conclusions, a data driven control strategy may be appropriate (see figure 3-1A). In the opposite case, a goal driven strategy may be appropriate (see figure 3-1B). However, it is often desirable to use some combination of these two strategies in order to maximize the efficiency of the problem solving procedure (see figure 3-1C). PROSPECTOR, for example, alternates between the two strategies. [14, p.23] It uses partial conclusions of its data driven phase to select a goal for its goal driven phase. The object is to converge on a rule which forms a link between the path from the initial state and the path from the goal state.

Yet another type of control strategy is the *blackboard approach*. A blackboard is a representation of the problem space itself. Each rule monitors a particular section of the problem space. It is applied when its antecedent appears

³ Synonyms for goal driven control are top-down, backward chaining and consequent reasoning.

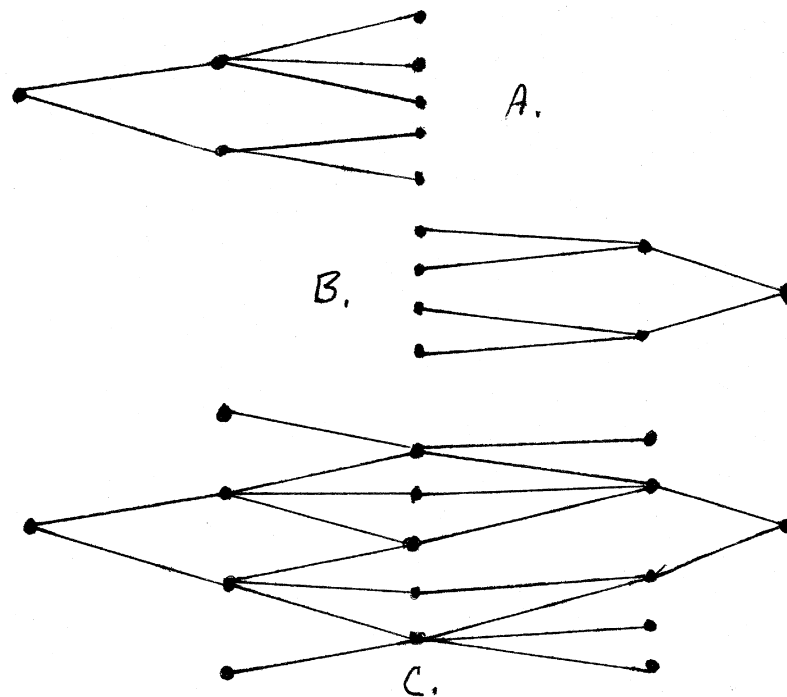


Figure 3-1: Control Strategies

in that section. [26, p.212] Its function may be to deposit information on the blackboard, to ask for input, or to produce output. As information is deposited on the blackboard, more rules are activated. The process continues until a goal is found, or the board becomes empty.

Data input to an expert system must be specified as true, false, numerical, unavailable, character strings, or as any *specific* type. Though natural language interfaces may be desirable in the presenting conclusions or explaining advice, possible errors in interpretation render it inadvisable to use as system input. For even in processing specified data, the system may not be provided with sufficient evidence to satisfy the antecedent of any rules. Two methods of dealing with this situation are calling subroutines to infer new evidence from existing data or asking

questions of the user to supplement the knowledge base. [47, p.84] It is desirable to use the first method first, in order to reduce the number of questions that must eventually be asked before a conclusion is reached.

One may object to the attribution of the title "expert" to a computer system because a machine lacks the categorical understanding which comes with long years of experience in a given domain. However, the basis for this objection is unclear at best, for it has been found that human experts can rarely document their reasoning processes, often resorting to after-the-fact justifications rather than coherent explanations. [24, p. 53] In addition, it has been found that expert judgement is highly dependent on the ability to recognise and match patterns of findings and hypotheses at a level of abstraction suited to a specific problem. The fact that most successful expert programs to date have relied heavily on a strict rule-based approach reflects the fact that they have all been analysis problems. But moderately successful expert systems exist for synthesis problems such as planning, fault repair, design and chemical synthesis. They also exist as interface programs for providing advice on the use of a complex system. [14, p.36] Future applications of expert systems may include localization problems. [18] They may even include problems of theory formation in indeterminate domains. [19]

This classic expert system paradigm is already undergoing change as knowledge engineers discover wider applications for automated expert advice. Models now exist which receive data not only from a single user, but from other computer programs. This makes it possible for expert systems to read data directly from complex measurement instruments and to analyze data immediately as it becomes available. These systems, not only suggest advice to the user, but also

monitor the user's actions. These actions may result in changed circumstances which provide the basis for new analysis of the constantly available data. Such a system can provide three types of advice: [47, p. 132]

1. Provide an interpretation of existing evidence and past action.
2. Offer advice on future action.
3. Check to ensure that current conclusions and actions are consistent with current available evidence.

Moreover, it is feasible that expert systems will acquire even more intelligent capabilities as progress is made in robotics, visual information processing, and machine learning.

CHAPTER 4

INTELLIGENCE

4.1. The Mind-Body Problem

It is evident from our introductory example that conversation with an expert system is not nearly fluent enough to merit attributions of personhood. Yet, it is not science fiction, but informed speculation, to venture that mechanical and perceptual interfaces may make expert systems capable of understanding significant portions of their environment. For a hypothetical, but not a farfetched example, a robot may be designed to monitor assembly line equipment and make minor adjustment if necessary. Given that knowledge is causally sustained belief, and that possession of conceptual structures constitutes belief, it seems obvious that such a mechanism possesses the necessary and sufficient conditions to *know* something about its environment. But can such a mechanism be considered to be intelligent?

If one believes that a person must have a soul in order to be intelligent, this rules out a machine. However, such an assertion may also rule out a human being. The *mind-body problem* is a label which refers to the set of theoretical issues which arise from a common sense distinction between mind and body. It can be viewed as an ontological issue which is based on the fact that there is no vocabulary to describe properties of the mind in purely physical terms. It can also be viewed as an explanatory issue, i.e., to scientifically understand the phenomenon

of mind. A *dualist* is someone who believes that mind and body are composed of two separate substances. The body is physical and the mind is non-physical. [1, p.15-36] This can be taken in one of three ways. The first is that the mind and the body do not interact. The second is that the body affects the mind, but not visa versa. But these conjectures tell us nothing about intelligence, for understanding and manipulating one's environment would then be properties of body alone. The third view on dualism is that of Descartes, wherein the mind and the body interact. Though popular in our culture, this argument has never overcome the objection that it is incoherent to suppose that a non-physical substance can affect a physical one.

The mind-body problem can perhaps best be dealt with Ockham's razor: *Do not posit plurality without necessity*. Another forceful argument against dualism asserts that thought is a property of organized matter, on par with impenetrability. [25, p.143-44] Mankind is most highly organized matter, therefore it is most intelligent. It is important to note in this sections that there could be no discussion of "artificial" intelligence without a rejection of the assertion that intelligence is a property of an immaterial mind.

4.2. Behaviorism

Nevertheless, intelligence is assumed to be a mental predicate. Its attribution is thus for the present subject to theoretical speculation and not empirical evidence. Unless, of course, one adheres to the doctrine that mental states are simply determined links in a causal chain between physical stimuli and behavior. There are two version to *behaviorism*. *Skinnerian behaviorists* maintain that an explanation of behavior is possible without reference to mental properties. [40] *Philosophical*

behaviorists define mental predicates by reference to a language of observation. [21] However, it may be impossible to define mental predicates in a way that does not include reference to other mental predicates. [39, p.93] Furthermore, philosophical behaviorism defines what *mental predicates* entail rather than defining *mental properties* themselves; it confuses the ontological issue in question with the related semantic issues.

Despite this rejection of Behaviorism, it is possible to use behavior as an indication of belief. Yet it is also possible that two systems have mental representations for concepts which nevertheless have different functional properties. A genuine concept must have both semantic and functional properties. The relationship between the two are the determinants of *intentionality*. Intentionality is that aspect of a concept which follows from general principles in semantic memory. It is easily understood as opposed to a concept's extension, which denotes the set of existing things to which the concept may be correctly applied. [42, p.10-11] A conceptual structure may embody a clear and distinct representation of water without carrying the information that water is equivalent to H_2O . Intentionality is particularly interesting in the account of knowledge as causally sustained belief because it allows for the possibility of forming a concept of something without recognizing its *essential* properties. A structure is *selectively* sensitive to the information characteristic of the learning situation. Semantic distinctions between structures of identical functional content render the conception of analytical truths a simple coding problem. Mental structures may thus be distinguished, or *type-individuated* by their functional properties alone. For instance, if the left-hand-side of two production rules had the same variable configuration (semantic

structure), then both rules may be applied. But two different semantic structures cannot exactly match a variable configuration of a single rule. It is the functional properties of conceptual structures which uniquely identify them. Therefore, even though a person's behavior indicates that he believes a liquid he calls "water" boils at 100 degrees celcius, it cannot legitimately be inferred that he believes " H_2O " boils at 100 degrees celcius.

4.3. The Identity Theory

The notion of a mental structure with semantic and functional properties is by no means new or unique to Dretske. Various *identity theories* make use of the notion that semantic properties and functional properties are one and the same. In other words, properties attributable to the mental life of an organism are identical to physical properties of the same organism. [41] There are two versions of the identity theory. The *token identity theory* simply asserts that every mental event, or kind is a physical event of some kind. Davidson outlines sketches the token identity theory as follows: [9, p. 88]

Although the position I describe denies that there are strict psycho-physical laws, it is consistent with the view that mental characteristics are in some sense dependent, or supervenient on physical characteristics...Dependence or supervenience of this kind does not entail reducibility through law or definition.

In contrast, *the type identity theory* asserts the existence of strict psycho-physical laws; that is, that for every mental property, or kind, there exists one physical kind which is identical with it. [38] The type identity theory is attractive because it implies that understanding of mind could compare to understanding of body. [16, p.6] But the type identity theory is untenable because evidence suggests that organisms of various species share mental properties, yet those properties must have different physical instantiations. [46]

4.4. Functionalism

The token identity theory is often associated with some brand of *functionalism*. Functionalism is the theory that every mental properties are type-individuated by their functional role in the behavior of the organism. [39, p.97] Unlike behaviorists and identity theorists, functionalists are not necessarily materialists. However, the attraction of functionalism lies in the fact that it renders feasible the notion of a purely physical mind. [36, p.175] Moreover, a dualist functionalist is bound to the implausible thesis that the brain alone cannot explain human behavior. [15, p.261] Thus, I confine my discussion of functionalism to *materialist functionalism*. Materialist functionalism combines the assertion that every mental event is identical to some physical event with the thesis that mental properties are type-individuated by their functional role in the behavior of the organism. Materialist functionalism avoids objections to the type identity theory because it allows mental properties with the same functional role to have a wide variety of physical instantiations. It also avoids objections to behaviorism because it defines mental properties rather than defining what is required for the attribution of a mental property.

However, objections to materialist functionalism do exist; the most forceful are *qualia objections*. Qualia is that aspect of human experience of which we are subjectively conscious, such as pain and appearances. Proponents of qualia insist that the standard for type-individuating mental states is subjective *felt experience*. [36] The *inverted qualia argument* asserts that the qualitative nature of experience in two different organisms may be functionally equivalent, but of distinct natural kinds. For example: two people see the same object. Both have been conditioned to call

it yellow, yet one *actually* sees red while the other sees blue. Then there is the *absent qualia argument*. This argument states that functional states of two organisms may be the same though one has felt qualia and the other does not. A popular example is that of Searle's Chinese room: a man is given a list of instructions for manipulating Chinese symbols; unknown to him, he is answering questions in Chinese. [37, pp.284-85] He experiences functional states without experiencing the corresponding qualia. An even clearer example for the absent qualia argument is that of an android which simulates human behavior but *feels* nothing. Functionalists have a ready reply to the inverted qualia argument. They may state that the laws of physics do not allow for human experiences to be qualitatively different. Or they may insist that the crucial aspect of qualia is not *felt quality*, but the functional relationships between mental structures. [6, p. 125] The second arguments may also be used against the absent qualia argument, in conjunction with the view that it is physically impossible for physical states to have motion without having some sort of felt qualia. [25] Still a third qualia objection to functionalism is that mental states are natural kinds whose essence is qualia. This argument is reminiscent of dualism. Materialist functionalists are committed to a thesis which implies that qualia are reducible to physical properties. These physical properties must therefore play functional roles attributable to felt qualia. Under this interpretation, qualia is that by which we identify a mental state and should not be confused with its essence.

Another challenging objection to functionalism has been that it is vague. Attempts to formalize functionalism have left its proponents with two separate theories. One may be called *intensional functionalism*. Intensional functionalism

holds that functional roles of mental structures must be specified by ascriptions of belief and desire. [10, ch. 1] To understand intensional functionalism, one must be able to view a system in terms of its physical components and also in terms of its design, or planned function; intensional functionalism is an account of how the design stance relates to the physical stance. This *intensional stance* assumes that a system's design is optimal given its goals; it thus utilizes a *rationality assumption* in ascribing belief and desire. The trouble with a rationality assumption is that to attribute a strict correlation between a system's goals and its design is to attribute, but not explain, intelligence. It is simply to assume that identical functions utilize identical mental structures. This view is therefore just as narrow as the type identity theory. Another significant problem with intensional functionalism is that humans are not ideally rational agents. [43, p. 48] They cannot be said to act in strict accordance with their beliefs and desires.⁴

The alternative clarification of functionalism is *Turing Machine Functionalism*. A Turing Machine is an abstract representation of a machine which solves problems by utilizing an *effective procedure*. [48, ch. 2] An effective procedure is a set of rules which define behavior step by step; an example is a search through a problem space. (see page 6) Turing Machine Functionalism is an attempt to specify what is meant by a 'functional role' with regard to a mental structure. It draws an analogy between mental processes and Turing-machine-computable functions. [32] The man who invented the Turing Machine concept, Alan Turing, wrote that if a computer could simulate human conversation well enough to fool an "interrogator"

⁴ Though intensional functionalism cannot adequately characterize intelligence, this term must not be confused with intentionality as defined above. Intentionality does have a place in clarifying the role of semantic structures.

50% of the time, then it must be considered to *think*. [45, p.12] Though this characterization of intelligence appears behavioristic, a Turing Machine Functionalist would assert that the attribution of intelligence is necessary upon encountering intelligent behavior. Block argues that the Turing test conception of intelligence is not a necessary, but a sufficient condition for intelligence. For it is implausible that intelligent behavior be produced by a non-intelligent system which simply *desires* to appear intelligent. [4, p.13] Furthermore, if a machine can use knowledge to perform functions which require intelligence if performed by a human, then its semantic structures must possess the functional properties necessary for intelligence as well. Turing Machine computability is a defense against those who view functional explanation as a purely hypothetical enterprise. [16]

CHAPTER 5

CONCLUSION

5.1. Objections

I have attempted to demonstrate that possession of mental structures with both semantic and functional properties is a necessary condition for manipulating knowledge and understanding one's environment. If a system possesses such structures and also exhibits intelligent behavior, then nothing prevents us from attributing intelligence to that system. However, objections to this view cannot be ignored. Rather, they must be aired and considered.

One objection to this conclusion may be that intelligence cannot be attributed to a program which is merely the creation of an intelligent programmer. But this argument cannot withstand the rebuttal that programmers themselves have been "programmed" by their upbringing and their education. [3, p.22] A computer's mental structures may mature more rapidly than those of a human, but they are nevertheless a property of the machine, and not a property of its programmer. Furthermore, an expert system may be prepared to deal with problems which the knowledge engineer had never even foreseen.

Another objection to the possibility of computer intelligence may be explained by the fact that people view computers as useful and productive machines.

Although true, this characterization glosses over significant differences between computers and other machines, even those which manipulate input and produce output. As an illustration of these differences, think of an assembly line machine which seals bottles with metal caps. Its inputs are filled bottles and metal caps. Its output is a sealed bottle. In performing its mechanical manipulations, the machine does not check to see if a bottle is filled before sealing it. It does not check to see that it is being fed a genuine metal cap rather than a piece of cardboard. It may perform manipulations on the wrong inputs or on no inputs at all. In contrast, a computer mediates its input and other symbols with its program. Symbols are strictly defined. They must be of a certain type. If symbols presented as input are not of the right type, a computer will not accept them. The machine will perform its manipulation on only those symbols it recognizes as valid input. Successful completion of symbol manipulation requires that all symbols conform to type-specifications at every stage of computation. If the type of any symbol clashes with the type required, no output is produced. (Confusion frustrates problem-solving activity.) This illustration highlights two attributes of computers which do not belong to other useful and productive machines.⁵

1. Type definitions provide knowledge of the *affective* capacity of a specified symbol.
2. Programs provide a characterization of the possible causal interactions of a given symbol.

Knowledge of both 1 and 2 is necessary and sufficient for *understanding* a specific symbol.

⁵ This illustration also highlights the fact that the analogue nature of a machine contributes nothing to its understanding of its environment; hence, it should not be considered a necessary condition for intelligence.

5.2. A Theoretical Obstacle

Even among those who do not object to the notion of computer understanding, there exists some disagreement on the level of intelligence so far achieved by research in AI. David Marr asserts that progress made in expert systems is misleading, for the development of intelligence must proceed at a much lower level of human cognition. [27, p. 20]

If one believes that the aim of information-processing studies is to formulate and understand particular information processing problems, then it is the structure of those problems that is central, not the mechanisms through which their solutions are implemented.

Marr views work in Artificial Intelligence at four different levels: basic circuitry analysis, functional architecture, algorithms, and *the theory of the problem to be solved*. He asserts that the last level ought to take precedence in AI research and that it is missing from research in production systems.

A contrasting view is held by Zenon Pylyshyn. He asserts that production systems may easily represent the functional architecture of the human mind. [33, p.74] He cites a number of aspects of production systems which he takes to significantly parallel human cognitive processes. These include: [33, p.81-82]

1. Focused attention on a limited number of symbols at a given time.
2. A data driven environment which accepts input from both internal and external sources.
3. Recognition of symbols which satisfy specific conditions.
4. The contents of the workspace includes all that occupies the system's attention (all partial paths toward possible goals).
5. Ease of expansion to include new "beliefs" and "goals".
6. The space-time tradeoff prompts grouping of data, which corresponds to the psychological phenomenon of "chunking" related bits of information.

Pylyshyn's view seems to coincide with that of Dretske in that information processing requires abstraction, that information is necessarily lost in the analogue-to-digital conversion characteristic of the cognitive operations involved in perception. He further asserts that most existing computational data structures (which he refers to as *semantic networks*) are suitable candidates for such internal descriptions. [34, p.177] In sharp contrast to David Marr, he maintains that: [33, p.85]

Any AI system is at some level a psychological theory, simply because the description of the intelligent task to which it is addressed already is a description of some psychological process.

5.3. Final Comments

Even strong advocates of computer intelligence must admit that the intelligence of existing AI systems is extremely limited by the problem domain. Understanding specified symbols is far less complicated than understanding an environment. It is the degree to which a machine understands its environment that is a measure of its intelligence. Thus the goal of Artificial Intelligence is to enable a computer to understand its environment. Because Artificial Intelligence researchers are steadily making progress toward that goal, at present there exists no barrier to machine intelligence. Conceptual barriers, however, continue to exist. They stem from the fact that computers are physically and mentally different from humans. Objections to the existence of computer intelligence were considered and dismissed by Alan Turing in 1950, yet the same arguments still receive attention today. [45] Though not to be ignored, theological arguments, arguments from consciousness, creativity and ESP are neatly avoided by the admission that computer intelligence is undoubtably qualitatively distinct from human intelligence. It is *intelligence* nonetheless. *Conceptual* barriers are no more compelling than their label implies. Inability to envision an intelligent computer does not negate the possibility of its existence.

References

- [1] Armstrong, D.M.
A Materialist Theory of Mind.
Humanities Press, New York, NY, 1968.
- [2] Ayer, A.J.
Knowing as Having the Right to be Sure.
In Roth, Michael D. and Leon Galis (editors), *Knowing, Essays in the Analysis of Knowledge*, pages 11-16. Random House, New York, NY, 1980.
- [3] Berkeley, Edmund C.
Ideas and Their Handling by a Computer.
Computers and People :21-24, May-June, 1981.
- [4] Block, Ned.
Psychologism and Behaviorism.
The Philosophical Review XC, January, 1981.
- [5] Boden, Margaret A.
Artificial Intelligence and Natural Man.
Basic Books, Inc., New York, NY, 1977.
- [6] Churchland, Paul M. and Patricia Smith.
Functionalism, Qualia and Intentionality.
Philosophical Topics XII:121-145, Spring, 1981.
- [7] Clancy, William J.
The Epistemology of a Rule-Based Expert System: A Framework for Explanation.
Technical Report STAN-CS-81-896, Stanford University, 1981.
- [8] Davidson, Donald.
The Material Mind.
In Haugeland, John (editor), *Mind Design*, pages 339-54. MIT Press, Cambridge, Massachusetts, 1981.
- [9] Davidson, Donald.
Mental Events.
In Foster, L. and Swanson, J.W. (editors), *Experience and Theory*. University of Massachusettes Press, 1970.
- [10] Dennet, Daniel C.
Brainstorms Philosophical Essays on Mind and Psychology.
TheMITPress", Cambridge, Massachusetts, 1981.
- [11] Drastel, GA and CA Kulikowski.
Knowledge-Based Rule Aquisition for Medical Diagnosis.
Technical Report CBM-TR-97, Rutgers University, 1982.

- [12] Dretske, Fred I.
Knowledge and the Flow of Information.
MIT Press, Cambridge, Massachusetts, 1981.
- [13] Dreyfus, Hubert L.
From Micro-Worlds to Knowledge Representation: AI at an Impasse.
In Haugeland, John (editor), *Mind Design*, pages 161-204. MIT Press,
Cambridge, Massachusetts, 1981.
- [14] Duda and Buchanan.
Principles of Rule-Based Expert Systems.
Technical Report STAN-CS-82-926, Stanford University, 1982.
- [15] Eshelman, Larry J.
Functionalism, Sensations and Materialism.
Canadian Journal of Philosophy 7(2):255-74, June, 1977.
- [16] Fodor, Jerry A.
Representations Philosophical Essays on the Foundations of Cognitive Science.
The MIT Press, Cambridge, Massachusetts, 1981.
- [17] Gettier, Edmund L.
Is Justified True Belief Knowledge?
In Roth, Michael D. and Leon Galis (editors), *Knowing, Essays in the Analysis of Knowledge*, pages 35-38. Random House, New York, NY, 1980.
- [18] Ginsberg, Allen.
Localization Problems and Expert Systems.
February, 1984.
Rutgers University.
- [19] Ginsberg, Allen.
A Model for Automated Theory Formation for Problem Solving Systems.
April, 1984.
Rutgers University.
- [20] Ginsberg, Allen and Sholom Weiss and Peter Politakis.
SEEK II: A Generalized Approach to Automatic Knowledge Base Refinement.
In *To be presented at the 9th Annual IJCAI Conference.* 1985.
- [21] Hempel, Carl G.
The Logical Analysis of Psychology.
In Block, Ned (editor), *Readings in the Philosophy of Psychology*, pages 14-24. Harvard University Press, Cambridge, Massachusetts, 1980.
- [22] Kastner, John Karl.
Strategies for Expert Consultation in Therapy Planning.
Technical Report CBM-TR-136, Rutgers University, 1983.

- [23] Kosslyn, Stephen.
Rutgers University.
Address at a conference: The Computer Metaphor of Mind, its Scope and Limits, April 22, 1983.
- [24] Kulikowski, Casmir A. and Sholom M. Weiss.
Representation of Expert Knowledge for Consultation.
In *Artificial Intelligence in Medicine*, pages 21-55. American Association for the Advancement of Science, Westview Press, Boulder, CO, 1982.
- [25] La Mettrie, Julien Offray De.
Man, A Machine.
Open Court Publishing Company, 1961 - 1748.
- [26] Lenat, Douglas B.
Computer Software for Intelligent Systems.
Scientific American :204-213, September, 1984.
- [27] Marr, David and H. Keith Nishihara.
Visual Information Processing, Artificial Intelligence and the Sensorium of Sight.
Technology Review :28-49, October, 1978.
- [28] McCarthy, John.
Epistemological Problems of Artificial Intelligence.
In Webber, Bonnie Lynn and Nils J. Wilsson (editors), *Readings in Artificial Intelligence*, pages 458-65. Tioga Pub. Co., Palo Alto, CA, 1981.
- [29] Minsky, Marvin.
A Frame-based Approach.
In Haugeland, John (editor), *Mind Design*. MIT Press, Cambridge, Massachusetts, 1981.
- [30] Newell, Allen and H. A. Simon.
Computer Science as Empirical Enquiry.
In Haugeland, John (editor), *Mind Design*, pages 35-66. MIT Press, Cambridge, Massachusetts, 1981.
- [31] Politakis, Peter and Shalom M. Weiss.
Using Empirical Analysis to Refine Expert System Knowledge Bases.
Artificial Intelligence 22:23-48, 1984.
- [32] Putnam, Hilary.
The Mental Life of Some Machines.
In Glover, Johnathan (editor), *The Philosophy of Mind*, chapter 6. Oxford University Press, 1976.

- [33] Pylyshyn, Zenon.
Complexity and the Study of Artificial Intelligence.
In Haugeland, John (editor), *Mind Design*, pages 67-94. MIT Press,
Cambridge, Massachusetts, 1981.
- [34] Pylyshyn, Zenon.
Imagery and Artificial Intelligence.
In Block, Ned (editor), *Readings in the Philosophy of Psychology*, pages
170-192. Harvard University Press, Cambridge, Massachusetts, 1981.
- [35] Rich, Elaine.
Artificial Intelligence.
McGraw-Hill, New York, NY, 1983.
- [36] Seager, William.
Functionalism, Qualia and Causation.
Mind 92:174-188, April, 1983.
- [37] Searle, John R.
Minds, Brains and Programs.
In Haugeland, John (editor), *Mind Design*, pages 282-306. MIT Press,
Cambridge, Massachusetts, 1981.
- [38] Shaffer, Jerome.
Mental Events and the Brain.
In Rosenthal, David M. (editor), *Materialism and the Mind-Body Problem*,
pages 67-72. Prentice-Hall, Inc., 1971.
- [39] Shoemaker, Sidney.
Some Varieties of Functionalism.
In Biro, J.I. and Robert W. Shahan (editors), *Mind, Brain and Function*, pages
93-119. University of Oklahoma Press, Norman, OK, 1981.
- [40] Skinner, B.F.
Selections from Science and Human Behavior.
In Block, Ned (editor), *Readings in the Philosophy of Psychology*, pages
37-47. Harvard University Press, Cambridge, Massachusetts, 1980.
- [41] Smart, J.J.C.
Sensations and Brain Processes.
In Rosenthal, David M. (editor), *Materialism and the Mind-Body Problem*,
pages 53-66. Prentice-Hall, Inc., 1971.
- [42] Sowa, J.F.
Conceptual Structures.
Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1984.

- [43] Stich, Stephen P.
Dennet on Intentional Systems.
In Biro, J.I. and Robert W. Shahan (editors), *Mind, Brain and Function*, pages 39-62. University of Oklahoma Press, Norman, OK, 1981.
- [44] Tannenbaum, .
Structured Computer Organization, 2nd Edition.
Prentice Hall, Inc., Englewood Cliffs, NJ, 1984.
- [45] Turing, Alan M.
Computing Machinery and Intelligence.
In *Computers and Thought*, pages 11-35. Robert E. Krieger Publishing Co., Inc., Malabar, FA, 1981.
- [46] Tye, Michael.
Functionalism and Type Physicalism.
Philosophical Studies 44:161-74, 1983.
- [47] Weiss, Shalom M. and Casmir A. Kulikowski.
Designing Expert Systems.
Rowman & Allenheld, Totowa, NJ, 1984.
- [48] Weizenbaum, J.
Computer Power and Human Reason.
W.H. Freeman and Co., San Francisco, CA, 1976.