

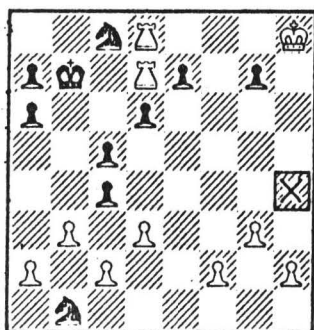
THE INTERACTION OF OBSERVATION AND INFERENCE IN A FORMAL REPRESENTATION SYSTEM

Robert Elliot Filman*
Artificial Intelligence Laboratory
Computer Science Department
Stanford University
Stanford, California

This work is an attempt to formally represent the knowledge required for the solution of a difficult retrograde chess problem (figure 1). This solution includes the extension of a formal deductive system to include an observational facility. Using a proof checker for first order logic, FOL [9], we have detailed a proof of the solution of the puzzle, including proofs for almost all of the necessary associated lemmas [2]. We shall highlight the various representational decisions made in the process of axiomatizing retrograde chess, discussing both the necessity for these particular choices, and their implications for designers of representations for other domains. This work is part of the search for *epistemologically effective* formalisms for artificial intelligence.

1. Introduction

Our consideration centers on a particular retrograde chess puzzle (figure 1). Its solution (from basic chess principles) is beyond the ability of any current computer program.



A piece has fallen off of the board from the square marked X. What piece was it? This position was achieved in a legal chess game, though there is no presumption that either player was playing to win.

figure 1

This problem was chosen because its solution "requires" both "deductive" and "observational" inferences, in a context isolated from other issues of correctness and sufficiency. Deductive inference is obtaining new proof steps by the application of syntactic inference rules. However, human reasoning proceeds not only by deduction, but also by the immediate recognition of results, a process we identify with *observation*. We have extended our representational system to include observational inference by performance of computation in a *semantic model*.

* Author's current address: Department of Computer Science, Indiana University, Bloomington, Indiana 47401.

** This work was supported in part by D-ARPA under contract MDA903-76-C-0206.

2. Inference

Human problem solving has several different modes. People are capable of *deduction*, applying syntactic rules to previous inferences. This has been the reasoning mode most frequently employed in A.I. programs. However, generally intelligent systems will need to reason by other schemes, such as *induction* (reasoning from particular cases to a general conclusion), *analogy* (modifying reasoning from a solution of another problem to fit the current situation), and what we call *observation* (quickly noticing an apparent conclusion, performed in our system by computation in a semantic model). If we are to create such programs, we must find a pragmatic representational mechanism, one that can support these other kinds of reasoning.

We perceive that the general A.I. problem is better approached by separating the representation of the knowledge of the problem domain (*epistemology*) from the search mechanisms used to manipulate that knowledge (*heuristics*). This distinction is discussed more completely in [7].

Classically, an axiom system is *epistemologically adequate* for solving some problem if there exists a solution of that problem from those axioms. For example, a world view based on interacting particles would be classically adequate. AI, intent on the production of intelligent programs, makes greater demands on its fundamental knowledge structures. We require not only the generality of adequate systems, but also economy of expression; that the solution be not only derivable, but that the derivation be short enough for a machine to actually produce it. We call this notion of both economical and adequate formalisms *epistemological effectiveness*.

This paper is primarily concerned with the qualities OF formalisms capable of epistemologically effectively representing both complex deductive and observational reasoning.

Deduction is obtaining conclusions by application of syntactic rules. The term *observation* is usually applied to human inference in several ways; they are unified by the notion of drawing a quick and immediate conclusion by examination.

Examples of human observation are seen in the technician reading the value of a meter, the mathematician recognizing the value of some simple function, and the clerk finding the amount of some transaction in a table.

This examination employs the following scheme: knowledge which is to be manipulated in deductive form is represented in an *extended first order logic* formalism. Associated with this logical system is a computational LISP model of these axioms. The results of the evaluation of functions on objects in this model are to be legal inferences in the deductive, logical system. This shall serve as our notion of *observation*. From the point of view of our intelligent computer, observation by computation in a semantic model is as opaque as sensory input.

3. Basic Formalisms

Just as our use of "English" is fundamental to our ability to present this paper, we have a fundamental need for an epistemological language for expressing reasoning. For that, we have chosen an *extended first order logic*. It is the usual predicate calculus, with the addition of *equality*, *functions*, *axiom schemata*, and inference by *computation in a semantic model*.

We use formal logic for several reasons: 1) The sentences of logic are unambiguous and easily understood. 2) First order logic has *explicit quantification*. 3) There has been much work in logic. Any system employing logic can avail itself of this work. Examples of such progress include resolution and decision procedures for propositionally logic. 4) Logic is a *general representation*. It is obviously not contrived for one particular domain. 5) Declarative representations in general, and first order logic in particular, are easily and uniformly *extendible*. They can be extended not only with new axioms, but also with new predicates, functions and individuals.

These, however, are the minor justifications for the use of logic. More importantly, 6) First order logic provides a precise sense of meaning. From the rules of inference inherent in the system it is clear what can be deduced from any axiom. 7) Almost all other current representations, be they microplanner, semantic nets, KRL, or whatnot, are variations on the rules of formal logic. They are all expressible in logical terms. These formalisms differ principally in their interaction with the heuristic system. They are all declarative systems, and (almost) equivalent in terms of epistemology. Reliance on pure logic has two positive attributes: we are isolating the epistemological issues from the implementation, and we are speaking the *lingua franca* of representation languages.

4. Proof Generation and Proof Checking

An intelligent computer program will need both an epistemologically effective world view and an appropriate set of heuristic procedures. This reasoning program is beyond our current abilities. However, we do not need this reasoning program to judge candidate epistemological formalisms. Rather, we need only to have our deductions certified. Clearly, the ability to accept a valid deduction is a prerequisite to the actual generation of that deduction. And an epistemologically adequate formalism can be tested by the use of a proof checker. This is essentially the *Missouri program* of McCarthy-Hayes [6], and is in the tradition of the *Advice Taker* [5].

We were fortunate to have available for this research the program FOL [19], an extended natural deduction proof checker for first order logic. FOL will act as our Missouri program, checking the validity of our candidate inferences.

FOL is a proof checker for an *extended sorted first order logic*. FOL has been extended to incorporate a *tautology decider* for propositional logic with equality. And, more importantly in FOL one has the ability to create a *partial semantic model* of one's world, and to evaluate the values of functions and predicates in that model, returning the results to the deduction level. This is called *semantic simplification*. This ability is satisfactory for simulating our notion of observation in chess.

5. Chess Puzzles

At the beginning of this paper, we mentioned a certain chess puzzle (figure 1).^{*} While space limitations preclude describing the solution of that puzzle, the reader should notice that, not only is the solution very complex, but that it also requires at least two different varieties of reasoning. Any solution will exhibit both *deduction*, inference of the form "Both sides can't be in check at the same time, black is in check, therefore white is not in check", and *observation*, inference of the form "I see black is in check". We perceive our task as finding the appropriate incorporation of this observational reasoning into the deductive framework.

If the reader himself solves the puzzle, he will surely perceive how complicated chess reasoning can be. In addition to complexity, chess (as a problem domain) has another appealing attribute: we can *unequivocally* solve this problem. This serves to help us to isolate the interesting features. And chess is a somewhat ill-structured domain, the result of historical development, rather than mathematical sparsity or toy simplicity. Our results, therefore, will principally be concerned with mechanisms, not representational details. However, understanding these mechanisms requires the presentation of those details.

We will employ two different forms of inference in our system, the standard *deductive* methods, based on axioms and natural deduction and augmented with decision procedures, and *observation*: semantic simplification performed by computation on a partial semantic model. We call this set of functions and data structures the *Chess Eye*. After defining a vocabulary of chess notions, we will consider the chess eye in greater detail.

6. Objects of the Chess World

Space limitations preclude detailing the entire chess axiomatization. Instead, we will describe only a few of the more important and interesting components.

The most obvious chess object is a chessboard, one of the *Boards*. Most chess problems are stated in terms of a board. Similarly, there are 64 *Squares* on any board, each with a unique identity, and 13 possible *Values*, such as *white pawn* and *empty*, corresponding to the objects on these squares.

However, values are different than the chesspieces themselves; the white queen's knight is not the same as the white king's knight; a promoted pawn is not the queen. Our reasoning will need these distinctions. Hence, we need a set of the 32 chessmen, called the *Pieces*.

The above, while somewhat sketchy, ought to appear fairly natural. However, as a representational mechanism, they are inadequate. We need to speak of the various things that must

^{*} The fallen piece in figure 1 was the white queen's bishop. The reader is referred to [2] for the detailed solution.

have happened to reach this point in the game, the capture that must have occurred, and so forth. Even the chess rules, in delimiting castling moves and draw conditions, refer to the entire game history. We therefore need an explicit set of historical states, the *Positions*, which can be thought of as the history (set of moves) employed in reaching this particular board. (There will be many different games that could reach any particular board.) Positions are a variety of state vector; the reader is referred to [6] for a more complete exposition on state vectors.

There is also the natural notion of the *Moves*, the explicit, discrete transitions between the positional states, and the *Colors*, black and white, associated with the two chess armies.

7. Predicates and Functions on Positions

Our most important manipulations will involve the state vectors, the positions. And the most fundamental relationship in reasoning about chess is that of legal move. Hence, we have the predicate *SUCCESSOR*(*p1*,*p2*) defined on two positions, if *p2* can be reached by a legal move from *p1*.

If *SUCCESSOR* is to be the succession function of our system, then, like arithmetic, we want a notion of "less than". This is called *PREDEGAME*. *PREDEGAME*(*p1*, *p2*) is true when *p1* occurred in the game that reached *p2*. Naturally, the initial position, *P0*, has the *PREDEGAME* relation to every legal game.

As each piece has its own identity, we can refer to the piece occupying a given square in a particular position. Thus, if *px* is a position that was played to reach the problem board, we have *Pos(px, BKRI) = WK*. (Or, on *BKRI* in *px* is the *WK*)

We also have functions which extract the last move from a position, and the actors used in that move (*Move*, *Mover*, *To*, *From*, *Taken*).

8. Predicates and Functions on Boards

Now, positions were only one variety of state vector. The other, perhaps more familiar, sort is that of *BOARDS*. Just as one can speak of the status of some square in some position, one can take the value on some square on a board (*VALUEON* (*GIVEN*, *BQ2*) = *RW*). (Note, however, that the range of this function is Values, not Pieces.)

The issue of legal moves is also important for boards; we have, for each piece, a predicate that determines the validity of any particular move for that piece on that board. Thus, for example, a bishop-valued piece can move from *BQ1* to *BKB3* on board *b* only if *DIAG(b,BQ1,BKB3)* is true on that board. We can take the composition of these various movement types, and talk about *MOVETO*(*b,v,sq1,sq2*) if, on board *b*, a piece of value *v* could move from square *sq1* to square *sq2*. However, legality of move is not purely a function of board; capture *en passant* and castling are not determined solely by board configuration.

Chessboards, as described in the previous section, will not even serve to represent the problem of figure 1. Consider the square with the *X* upon it. It has no representation within the system of ordinary values that we have described. We resolve this difficulty by the introduction of a new constant, of type *Values*, one that stands for "I don't know what's here, but it is something." This is the constant *UD* (for "undefined" value.) Permitting an undefined value to be on the squares of boards creates a natural ordering on boards by greater definition. We call this ordering *SUBBOARD*(*b1*,*b2*).

We need to tie boards and positions together. Obviously, a board with no undefined squares is the result of playing out a position. The function that extracts this board is *Tboard*. We state that $\forall p \ b.(BOARD(p,b) = SUBBOARD(b,Tboard(p)))$. In reasoning about chess puzzles, we typically take the problem board, and infer properties of any position that has the *BOARD* relation to it.

9. Observation in the Chess World

In our definition of the FOL inference system, we allowed computation in a partial semantic model to be a legal inference. We therefore define a LISP representation for (some) of the objects of our chess world, and to declare LISP functions to compute the computable predicates and functions on that model. For example, in our LISP model, atomic individual constants, like the black king and the white queen's bishop are LISP atoms; chessboards are eight long lists of eight atomic values.

Attachments are then made to functions and predicates such as *VALUEON*, *ORTHO*, and *MOVETO*. For example, the predicate *MOVETO*, which is defined axiomatically as:

```
Vb v sq1 sq2.(MOVETO (b,v,sq1,sq2) =
  ((VALUER(v)^(ORTHO(b,sq1,sq2))^(VALUEB(v)^(DIAG(b,sq1,sq2)))v
  (VALUEQ(v)^(ORTHO(b,sq1,sq2)) v(VALUEQ(v)^(DIAG(b,sq1,sq2)))v
  (VALUEK(v)^(KINGMOVE(sq1,sq2)))v
  (VALUEN(v)^(KNIGHTMOVE(sq1,sq2)))v
  (VALUEP(v)^(PAWNMOVE(b,v,sq1,sq2)))))
```

is attached to the function:

```
MOVETO (LAMBDA (b v sq1 sq2)(COND
  ((VALUER v)(ORTHO b sq1 sq2))(VALUEB v)(DIAG b sq1 sq2))
  ((VALUEQ v)(OR (ORTHO b sq1 sq2)(DIAG b sq1 sq2)))
  ((VALUEK v)(KINGMOVE sq1 sq2))
  ((VALUEN v)(KNIGHTMOVE sq1 sq2))
  ((VALUEP v)(PAWNMOVE b v sq1 sq2))));
```

in the semantic model. We build up our hierarchy to have predicates such as *BLACKINCHECK* (which determines if black is in check on a given board), computable in the chess model. Thus, though our system is capable of expressing complex statements about chess, it is still able to derive the fact that black is in check on the problem board in a single inference.

The semantic simplification mechanism also has the ability to check WFF's quantified over finite sets by case computation in the semantic model. For example, the theorem: $\forall x. \exists sq. Pos(P0,sq) = x$ (every chesspiece was on some square in the initial position) required 165 steps to derive deductively, but only a single semantic simplification.

This use of procedural computation in a model is a form of observation; the intelligent program, whose internal language is the formal predicate logic and axioms, can use the same mechanism for simplification done in its semantic model and observed events in the real world. What we are providing here is a mechanism for perception; that its primary use in computers will be a form of internal visualization or hallucination is no more a problem than accepting the human use of internal visualization in human problem solving.

10. Axioms

The most important axioms in this system are those that delimit legal moves. For example, consider the following representative axiom, *MCONSEQA* (move consequences A):

axiom MCONSEQA: $\forall r, q. (\text{SUCCESSOR}(r, q) \supset$
 $((\neg \text{WHITETURN } r \supset \text{WHITETURN } q) \wedge \neg \text{POSITIONINCHECK}(q, \text{Color } r) \wedge$
 $\text{Prevpos } q = r \wedge (\text{WHITEPIECE } \text{Mover Move } q \supset \text{WHITETURN } r) \wedge$
 $\text{Pos}(r, \text{From Move } q) = \text{Mover Move } q \wedge \text{Pos}(q, \text{To Move } q) =$
 $\text{Mover Move } q \wedge \text{Pos}(q, \text{From Move } q) = \text{EMPTY} \wedge$
 $(\text{CAPTURE } \text{Move } q \supset \text{Pos}(r, \text{To Move } q) = \text{Taken Move } q) \wedge$
 $(\text{CASTLING}(r, q) \vee \text{EN_PASSANT}(r, q) \vee \text{SIMPLELEGALMOVE}(r, q)))$;

It states some conditions on legal moves; which side is on move, not leaving one's king in check, where the pieces in the move are, and the beginning of a taxonomy of move types.

It is also important to talk both about what remains true between moves. For example, those squares not involved in the move retain the same contents between any position and its successor.

axiom MCONSEQD:
 $\forall r, q, sq. ((\text{SUCCESSOR}(r, q) \wedge \neg sq = \text{From Move } q \wedge \neg sq = \text{To Move } q \wedge$
 $\neg (\text{CASTLE } \text{Move } q \wedge \neg sq = \text{Alsofrom Move } q \vee sq = \text{Also to Move } q)) \wedge$
 $\neg (\text{ENPASSANT } \text{Move } q \wedge \neg sq = \text{Takenon Move } q)) \supset$
 $\text{Pos}(r, sq) = \text{Pos}(q, sq))$;

11. Chess Induction

Another feature of the chess axiomatization that deserves mention, is *Chess Induction*. Chess induction is an axiom schema that states that if a property $\mathcal{P}$ is true of a position p , and remains true over the successor relation, then it will be true in all descendants of p (all games that can be reached from p).

We usually use the initial position, $P0$, for p , and prove properties true of all legal positions. More formal, this is:

$$(\mathcal{P}(P0) \wedge \forall r, p. ((\mathcal{P}(r) \wedge \text{SUCCESSOR}(r, p)) \supset \mathcal{P}(p))) \supset \forall r. \mathcal{P}(r)$$

Chess induction is a useful deductive method. For example, using chess induction, we prove theorems such as: *A piece not on its original square has moved in this game (or been the moving rook of a castle)*:

$$\forall r, sq, x. ((\text{Pos}(r, sq) = x \wedge \neg (\text{Pos}(P0, sq) = x)) \supset \exists q. ($$

 $(\text{PREDEGAME}(q, r) \vee q = r) \wedge ((\text{Mover Move } q = x \wedge \text{To Move } q = sq) \vee$
 $(\text{CASTLE } \text{Move } q \wedge \text{Alsofrom Move } q = x \wedge \text{Also to Move } q = sq)))$

And that *bishops stay on a single color square*:

$$\forall r, sq1, sq2, ybi. ((\text{Pos}(P0, sq1) = ybi \wedge \text{Pos}(r, sq2) = ybi) \supset$$

 $(\text{WHITESQUARES}(sq1) = \text{WHITESQUARES}(sq2)))$

This is a powerful technique for dealing with the frame problem. If we have control over the types of interactions available in state transitions, induction schemas such as this one allow us to prove constant properties of distant states.

12. The Proof

This has been just a small sprinkling of examples from the chess axiomatization. We hope it has been useful in conveying their flavor and form. From this axiomatization, we preceded to generate a proof of the solution to the chess puzzle.

The proof itself is too long to detail, but we are able to present a few comments about its "shape". The form of the proof matched, fairly closely, an English explanation of the detailed solution to the same problem. The human proof required 68 "steps"; the FOL "proof", 405. In the process of deriving the FOL proof, 159 general chess lemmas and theorems were

proven. These required 1702 steps. Additionally, six lemmas specific to this problem were also demonstrated, requiring another 136 steps. About half of the proof steps in the main proof were instantiations of axioms and theorems, and another quarter were requests for the confirmation of the chess eye. The complete FOL proof may be found in [2].

13. Syntactic and Semantic Reasoning

We will attempt to convey some of the epistemological design concepts uncovered in this examination. This is a brief description; a more thorough examination may be found in [2].

One of the major distinctions underlying this work is that knowledge can be successfully modeled in two different forms. One is by the application of *syntactic rules* and the other, *model based semantic computation*. From the perspective of the person communicating with a representational system, syntactic mechanisms convey by their *form* the information (and possible uses of that information) they represent. Thus, in our system the *axiomatic* knowledge as syntactic. Knowing the rules of inference, it is clear what is derivable from any axiom.

Semantic rules, on the other hand, are magic. We are not told the justification for any rule, why it works, how it works, only that we are to accept its results. Typically, it will be performing this action by functional evaluation in some partial model of the domain, complete with hidden assumptions about what is right and possible there. Such inference is performed in this system by the functions of the Chess Eye.

Detailed exploration of formal chess reasoning reveals that it is a fallacy to believe a *simple methodology* will be able to express the knowledge of a complex domain. Expressing complex relationships will require complex linguistic machinery. For example, any system dependent upon viewing only a single model, updated with each action, will be unable to compare states resulting from different actions in anything but the most primitive way. Any system that thinks that a "shortest path net traversal" algorithm will permit it to avoid dealing with inconsistency is going to be unable to do deep reasoning that relies on disjunction or case analysis.

14. State Vectors

The manipulation of *state vectors* played the central role in this axiomatization. State vectors are like the practice of proofs in formal logic. Everyone knows how they can be used, but no one is every caught manipulating one in a complicated way.

We remind the reader that there were two flavors of state vector employed in this research; the first is the *static* vector, the boards. Boards were used to display a particular *slice* of time. Similarly, these chessboards were a fully realized, concrete object upon which calculations could be performed. The second is the *historical* vector, the position. A position contains the complete history of any particular game. However, it has no explicit representation in the model.

Each of these had important application. The full realization of the board (the association with any board with a data structure that completely defined that board) permitted computation on that board; and this employment of the Chess Eye proved remarkably useful in reducing the magnitude of derivation. However, the historical state vector, the position, was also vital. Much of the work of this proof consisted of comparing states, asserting that a particular state with

particular properties must have existed. Usually this state occurred in the game played to reach some other state, and was interesting for that reason.

There is a third kind of state vector, the *possible world*. This vector contains information not only about the status of objects in the world, but also propositions about the propositions about those worlds. The most common use of these possible worlds is in semantic modeling of model logic systems. Possible worlds were not needed to model the simple world of retrograde chess.

State vectors seem to be one of three possible approaches to the problem of extending the set of things one can talk about in a formal system. This is the method of *reification*, or the making of new objects. Here we postulate new, "ordinary" objects, from which propositional information can be obtained by use of the axioms. These objects can be manipulated by the use of the ordinary rules of the standard logic.

A second possible alternative is provided by *modal logic* formalisms; these attempt to talk about propositions by adding new "quantifiers" on propositions, and new inference rules to the deductive system. Unfortunately, if the proposition to be talked about contains several possible modals, the system can become rather cluttered with additional inference rules. And each additional modal requires extending not only the inference system, but also the heuristic system.

The third alternative focuses on the notion of *meta level reasoning*. That is, the propositions on one level become the objects of the next level. An additional set of inference rules is required for transferring between these levels. Since this scheme can be recursively applied, no additional mechanisms need be added to handle other proposition types. This is essentially a more advanced form of reification.

Of these three, the third seems most promising, and it is currently being incorporated into FOL. If this axiomatization were to be repeated, the features of this meta level reasoning would have allowed more "incompletion" in the manipulation of the semantic model. However, the state of FOL at the time this research was performed compelled the use of reification and state vectors rather than this last, more general scheme. These distinctions are discussed more fully in [10].

One other major success in the use of state vectors ought to be mentioned: we had great success applying induction schema to historical vectors to prove properties of distant states. The induction schema *Chess Induction* stated that if a property was true over the successor relation, then it held in all descendants of any state for which the property is true. Chess induction permitted the proof of many previously intransient theorems, such as "Only pawns can promote" and "Every piece now on some square either moved there in a previous position, or started on that square". We believe that this induction schema provided a good partial remedy to the frames problem, permitting proofs about distant states without even the trouble of simulating the actions involved.

15. Compound Objects

The state vectors are examples of another form of reification, that of creating *compound objects*. The mathematical comprehension principle, creating sets, lists and bags, are forms of compound objects. Our experience is that compound objects play a critical role in even the simplest epistemology.

Compound objects are related to the idea of *partially defined objects*. A partially defined object is one that is a representation for something we have conceptually more information. We suggest there are two different varieties of partially defined objects that need to be dealt with in a complex reasoning system. One of these is exhibited by the *boards* with their explicit use of *UD*, the "undefined" constant. A board with an undefined in it is not conceptually a first class object of our system; rather, it is an explicit representation an object about which we only have partial knowledge. This is its tremendous advantage; with this explicit representation, we can use a partial board as data for a functional evaluation. However, the use of this explicit undefined limits the set of things that can be unknown about a partial board; we might know that an undefined square contains some officer, or that there is a queen somewhere on the board, or that this board was a legal move away from some other board. None of these partially defined objects is representable by this scheme. Each of them could be represented by some other scheme, *but not all such partial information* can be represented by any scheme that does not represent its information propositionally. Representing propositional information in the model is already meta-reasoning [10].

The alternate variety of partially defined object exhibited in this representational system is the *position*. In one sense, the position is completely defined; the position does not have any formal elimination of information. On the other hand, except for the most trivial positions, one knows nothing about any position that cannot be established propositionally. It is only possible to infer properties of a position from its relation with other positions and boards.

We see then that the board is the *concrete* realization of the *abstract position*. Any epistemologically effective system is going to need to deal with both abstract objects (for which there can be no representation in its model) and concretions of those objects (for its model to manipulate).

This discussion leads us to touch upon the issue of *multiple representations* or the different *aspects* of similar objects. What we are considering here is the ability to refer to objects that have a common conceptual source in different fashions.

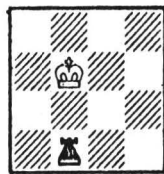
We see this division in our chess axioms. Inherently, there is only one conceptual object associated with any square at any point. Nevertheless, we find the necessity for dealing with this object in two different garbs: piece and value. We essentially have things to say about the pieces that both abstract information out (it's a white rook, not the white queen's rook) and that add information in (but it used to be a pawn). We see this same division in the board/position duality, where the board is really a "slice" of a position.

16. Linguistic Generality

This final section provides a perspective on how well we've provided a language for reasoning about chess. We need to consider a two part perspective: What were the failures in this representation system? How well could it be used to talk about problems other than "A piece fell off the board ...?"

There are two important linguistically failings in this axiomatization. The first of these was the failure to include general notions of *compound objects*, particularly sets and their kin. It was possible to deal with arguments equivalent to the "pigeon-hole principle" by case checking; but these are not naturally case arguments.

The other source of regret is that the representation system, particularly in the semantic model, is tied too closely to the rules of legal chess, and too closely to too simple an idea of representing chess knowledge. This point can be better illustrated with a pair of examples. Consider this problem:

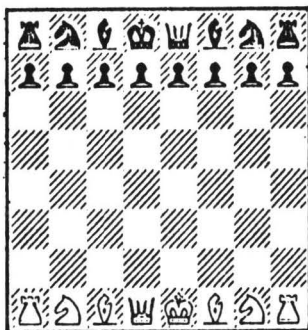


Is white in check?

figure 2

The current axiomatization, as currently formulated, would not recognize this "fragment" as a piece of a "board". However, this is a natural extension for the human problem solver. The axiomatization can be criticized for adhering too closely to the rules of chess, and ignoring its "primitives". Correction of the axiomatization to reflect this more primitive level would not need to be a major revision.

This second example displays a more fundamental linguistic difficulty. Consider this problem, from [3], somewhat more conventional in its phrasing.



White to play and mate in four.

figure 3

This is not so much an issue of mating in four, so much as recognizing that (as the black king and queen are on the wrong color squares) the pieces have exchanged sides. While the current axiomatization could be used to prove that if white started at the bottom of this board, then this board could not have been reached in a legal game, in some sense, the puzzle needs to be solved before it can be put in the acceptable form for the axioms. What we have here is an issue of language. There are certainly many questions one would like to ask in, for example, English, for which natural language is inadequate. Imagine, if you will, the circumlocutions involved in giving a written spelling test. The moral here is an epistemological incompleteness theorem; there will always be things that cannot be said in any system.

We have considered the axiom system in the light of other retrograde problems, and find that it is effectively capable of expressing their solution. In particular, we have considered the problem in [4],* and find no difficulty in expressing its solution in our axiomatization.

17. Summary

We have taken a difficult problem of retrograde analysis chess, detailed a set of axioms for the rules of chess, and have shown that those axioms, together with a clever natural deduction system and the ability to simplify in a semantic model, are sufficient to produce a manageable derivation of the solution of a complicated problem. We have considered some of the conclusions that can be formed about formal knowledge representations from this work. Particularly we have explored, the nature of the objects of the chess world, such as state vectors, concrete and abstract objects, and compound objects and the value of things such as semantic simplification, meta reasoning, modal reasoning and induction schema.

18. Acknowledgments

I would like to acknowledge the helpful advice provided by Drs. John McCarthy, Richard Weyhrauch and Terry Winograd in the course of pursuing this research.

19. References

- 1 Filman, Robert E., and R.W.Weyhrauch; An FOL Primer. Memo AIM-288, Stanford University, September 1976.
- 2 Filman, Robert E.; The Interaction of Observation and Inference. PhD thesis, Computer Science Dept. Memo AIM-327, Stanford University, March 1979.
- 3 Gardner, Martin; Mathematical Games in the Scientific American 202:5 (May 1959).
- 4 Gardner, Martin Mathematical Games in the Scientific American 228:5 (May 1973).
- 5 McCarthy, John; The Advice Taker. In M. Minsky, ed. Semantic Information Processing, MIT Press, Cambridge, Mass. 1968.
- 6 McCarthy, John, and P.J. Hayes; Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Machine Intelligence 4, B. Meltzer & D. Michie, eds. Edinburgh University Press, Edinburgh, 1969.
- 7 McCarthy, John; Epistemological Problems of Artificial Intelligence. In Proc. IJCAI-77, MIT, Cambridge, Mass., August, 1977, pp 1038-1044.
- 8 McCarthy, John; Concept valued functions. To appear in Machine Intelligence 9, D. Michie, ed.
- 9 Weyhrauch, Richard W.; A Users Manual for FOL. Memo AIM-235.1, Stanford University, 1977.
- 10 Weyhrauch, Richard W. and Robert E Filman; Meta Reasoning and Extended Inference for AI (in preparation).

* Black king on WQ1, bishop on WQ5, rook on QN5; white bishop on WQR4. Where is the white king?