An Example of Human Chess Play in the Light of Chess Playing Programs

Allen Newell and Herbert A. Simon
August, 1964

This paper is to appear in a volume honoring Norbert Wiener. It may not be used or reproduced without the permission of the authors. This research was supported in part by Research Grant MH-07722-01 from the National Institutes of Health and in part from contract SD-146 from the Advanced Research Projects Agency.

Carnegie Institute of Technology

An Example of Human Chess Play in the Light of Chess Playing Programs[1]

Allen Newell and Herbert A. Simon

Game playing machines have occupied a prominent role in the history of cybernetics, standing as easily understood examples of artificial intelligence. Chess has a favored place in these discussions, no doubt, because of its reputation in the West as a pure contest of intellect. Wiener himself devoted a small appendix in Cybernetics to observing how a chess automaton might be built, and he returned to the theme again and again in later writings and talks. Gradually the examples shifted from chess programs to checker programs, due to the phenomenal effectiveness of Samuel's checker program, and its ability to improve its performance from experience. In all of these discussions Wiener's immediate object was to establish the intellectual power of machines as a premise for his social criticism and concern (to use his own words) with the human use of human beings.

A discussion of chess playing could follow this line of analysis -- could concentrate on the current status of chess programs and their implications for whether machines do or do not pose a threat to the evolution of our society. But the study of chess programs can serve other interests; for example, interests in human psychology. How do humans in fact play chess or, more generally, how do they think and reason? Wiener's own interest in game playing programs apparently never had this motivation, but he was deeply interested in related questions at a more physiological level. He was hopeful

[1] We would like to acknowledge our indebtedness to Harvey E. Wagner, who obtained and made a preliminary analysis of the protocol discussed in this paper. This research reported here was supported in part by Research Grant MH-07722-01 from the National Institutes of Health and in part from Contract SD-146 from the Advanced Research Projects Agency.

that at many points an understanding of artificial mechanisms would yield insight and guidance in problems in human physiology.

This paper is concerned with the use of chess programs to study human thinking. The work on chess programs has produced a collection of mechanisms sufficient to play chess of modest caliber. Independently of their detailed characteristics, they help understand what must be done in order to play chess. Other problem solving programs also contribute to this understanding, but for convenience we will restrict our attention to chess programs, viewed as tools of analysis to explore how humans play chess. Since considerable previous analysis already provides substantial evidence of the fruitfulness of these tools, we will say little about the relation of models, theories and simulations to the things they talk of (Green, 1963; Newell and Simon, 1963).

Our approach will be to examine in some detail the behavior of a man deciding what move to make in a specific middle game position. We have available a protocol, a transcript of the verbal behavior of the man while he is analysing the board and making his decision. Previous work with protocols in other tasks (proving theorems, guessing sequences, learning concepts) has aimed at constructing computer programs that match the behavior in detail. In this paper we will undertake only the first stages of such an analysis, laying bare the reasoning the subject employed, by examining his protocol in detail. The analysis will draw upon our general knowledge about reasoning mechanisms and how to organize information processing.

We will first summarize briefly what has been learned to date from work on chess programs. Then we will discuss human performance in chess. With these preliminaries out of the way we will devote the remainder of the paper to the analysis of the protocol.

Chess Playing by Programs

Basic approach. The fundamental scheme used for playing chess by computers was introduced very early. In 1948 Wiener described the scheme thus:

> I think it possible to construct a relatively crude but not altogether trivial apparatus for this purpose [i.e., to play chess]. The machine must actually play -- at high speed if possible -- all its own admissible moves and all the opponent's admissible ripostes for two or three moves ahead. To each sequence of moves it should assign a certain conventional valuation. Here, to checkmate the opponent receives the highest valuation at each stage, to be checkmated, the lowest; while losing pieces, taking opponent's pieces, checking, and other recognizable situations should receive valuations not too remote from those which good players would assign them. The first of an entire sequence of moves should receive a valuation much as von Neumann's theory would assign it. At the stage at which the machine is to play once and the opponent once, the valuation of a play by the machine is the minimum valuation of the situation after the opponent has made all possible plays. At the stage where the machine is to play twice and the opponent twice, the valuation of a play by the machine is the minimum with respect to the opponent's first play of the maximum valuation of the plays by the machine at the stage when there is

> only one play of the opponent and one by the machine to follow. This process can be extended to the case when each player makes three plays, and so on. Then the machine chooses any one of the plays giving the maximum valuation for the stage n plays ahead, where n has some value on which the designer of the machine has decided. This it makes as its definitive play.
>
> Such a machine would not only play legal chess, but a chess not so manifestly bad as to be ridiculous...(Wiener, 1948, pp. 193-194)

This set of ideas, as the paragraph indicates, is based on a game theoretic analysis in the style of von Neumann and Morgenstern. The matter was more thoroughly explored, although still in an essentially discursive vein, by Shannon in 1950 (Shannon, 1950). It was not until 1957 that the first chess program was constructed (ignoring some hand simulations and some programs for playing end games) (Kister, et al, 1957). Several programs were constructed between 1957 and 1959 (Bernstein, et al, 1958; Newell, Shaw and Simon, 1958) after which a period of relative quiescence ensued. Recently, however, a few new programs have been put into operation. One of these, constructed at MIT, is probably the best chess playing program to date (Kotok, 1962). It has done well on occasion against quite good players when given such odds as Queen or Rook.

It has been noted many times that when chess is viewed in game theoretic terms, it consists of an exponentially expanding tree of consequences, which has about 30 branches per node and runs about 40 moves (80 half moves) deep. This results in some 10^{120} end points to be examined, if the whole tree is to be searched. The problem faced by all the chess programs (and they have all used the growing tree as a framework) is how to reduce the number of positions examined to some reasonable size. As the quotation from Wiener indicates, the original ideas were fairly simple: terminate the search at some depth, n, and substitute for the unknown value of the position a score that seems reasonable in the light of chess knowledge[2]. One early program (Kister, et al, 1957) played in essentially this fashion, using n to be four half moves (which leads to about 10^6 positions considered in total).

Mechanisms. Generally, the chess programs have developed in the direction of making as many aspects of the situation as possible variable -- what moves are to be considered, how deep to search, etc.. Various rules (called heuristics) are used to determine these variables as a function of the particular chess position and the need to limit search. We can distill the essence of these programs in the following collection of mechanisms (not all of which are used in any one program):

---

[2] In checkers it has proved possible to have the program optimize its scoring function both by play and by analysing recorded master games (Samuel, 1959); however, this has not been tried seriously in chess.

- Plausible move generators. Given a position these generators produce moves that are appropriate in terms of standard chess theory.

  Example: Generate all moves that add a defender to an attacked man.

  Example: Generate all moves that occupy a weak square of the opponent ("weak square" is a standard chess concept).

- Considered-moves rule. This rule chooses various of the plausible move generators, considering them in some order, until a sufficient number of moves have been obtained.

  Example: Take the first 7 plausible moves.

  Example: Take all the moves the plausible move generators can suggest.

  Example: Generate all moves according to some plausible move generators, and keep only those that are not "foolish" -- say, that do not leave the man en prise.

- Static evaluation. This routine assigns to a position an overall value or score.

  Example: Let S = (value of men on board, taking Q = 9, ..., P = 1) + (total squares men can move to); then static value = S(White)/S(Black)

  Example: In addition to S above, assign points for the safety of the King position, isolated pawns, weak squares, open files, etc., etc., etc.

- Static position test. If a position is static, then it can be assigned the static value. If it is not static, the consequences to be found by making the considered moves must be examined (the position is then called dynamic).

  Example: If the move is N-deep, it is static; if it is less than N-deep, it is dynamic.

  Example: If capture is possible, the position is dynamic; otherwise it is static.

- Value inference procedure. This determines the value of a dynamic position from the values of its consequences. (This applies only to dynamic positions, since static positions already have values assigned.)

  Example: (Minimax): The score shall be the best attainable according to the interests of the side which can choose the move. If points are counted with respect to White, this involves maximizing for White and minimizing for Black.

- Move selection rule. This chooses the move to be played from the given position, as a function of the values of the considered moves.

  Example: (Choose the move with the best value either maximum or minimum as the case may be).

  Example: Choose the first move that attains a value better than an aspiration level (either greater or less as the case may be).

Programs constructed in the spirit of the scheme presented above do not eliminate the exponential growth of the tree of consequences; they only serve to control it. Exploration ranges from about 50 positions up to about 800,000 depending on the heuristics. Figure 1 shows an exploration tree from one of our own programs (most of the other programs explore too many positions to permit human examination of the search tree). In Figure 1 the positions considered are indicated by small circles, the initial position being in the upper left corner. The moves considered are the branches extending to the right and below; each is labeled with the move[3].

---

[3] Standard English chess notation is used, except that Black men are primed to permit identification in contexts where the side is not apparent.

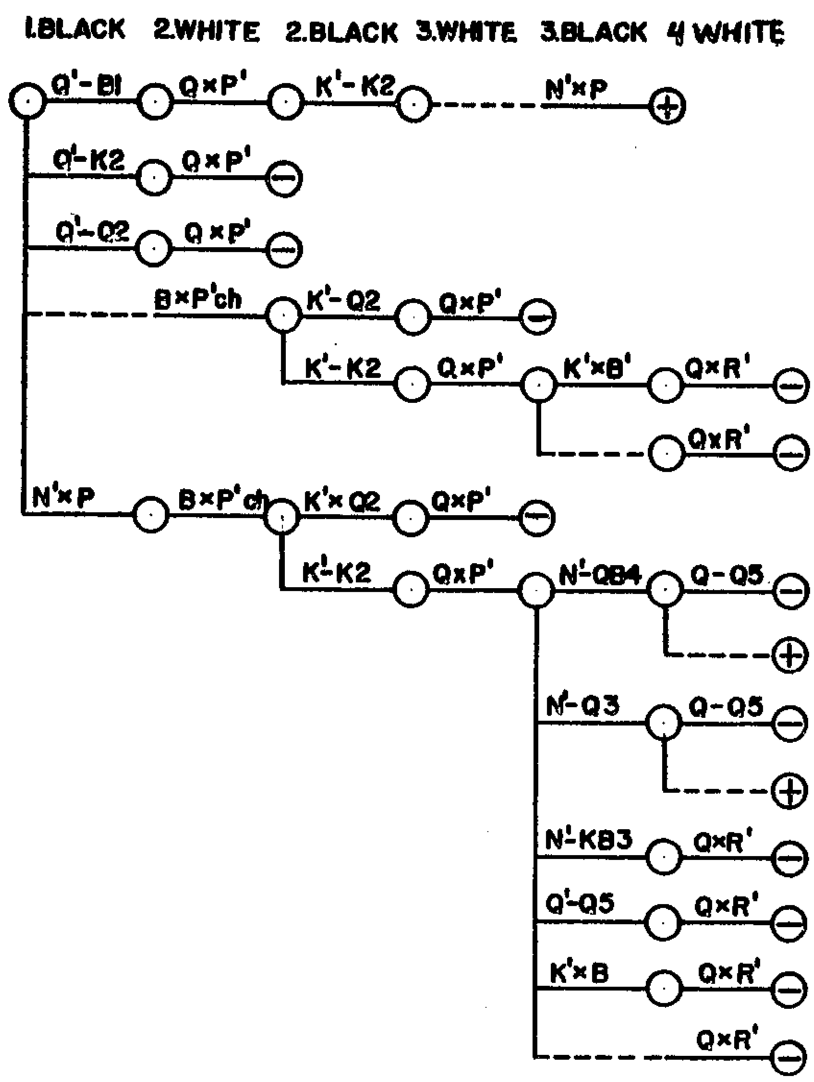

Figure 1: Exploration tree of chess program

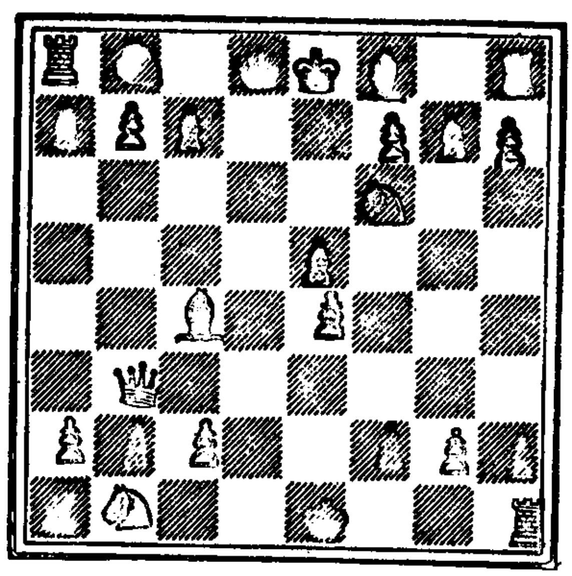

Figure 1 (continued)

The dashed branches indicate that no move was generated, but that analysis was continued with the next move of the opponent. Thus in Figure 1 five initial moves were considered by Black, three with the Q', one "No-move" and one with the N'. The final positions are marked with + or -, which is the relative evaluation for the side making the move -- i.e., Black.

All of the specific knowledge about the game of chess is buried inside the parts of the scheme -- in the programs that determine what moves are plausible, in the static evaluation function that specifies how much a piece is worth relatively speaking, and so on. The scheme itself is quite general. Similar schemes of selective search are used in most of the other problem solving and game playing programs, such as the checker programs and the theorem proving programs. The general conclusion from all of these is that heuristic search (that is, search under the control of rules that prune and shape the growing tree of possibilities) is sufficiently powerful to produce problem solving at a level that is interesting by human standards.

Chess mating program. The efficacy of heuristic search techniques has been further verified in a somewhat special but more demanding part of chess -- the discovery and verification of mating combinations. One aspect of good human play is the ability to discover mates that are up to eight moves or even more into the future, (over sixteen half-moves, in terms of the expanding tree). With this depth any exponential expansion generates huge numbers of positions to be examined; consequently, the mating combination task affords a nice test of how powerful these problem solving techniques are. Figure 2 shows and example of a computer program for discovering mating combinations at work on a position that arose in a game of Ed. Lasker v.

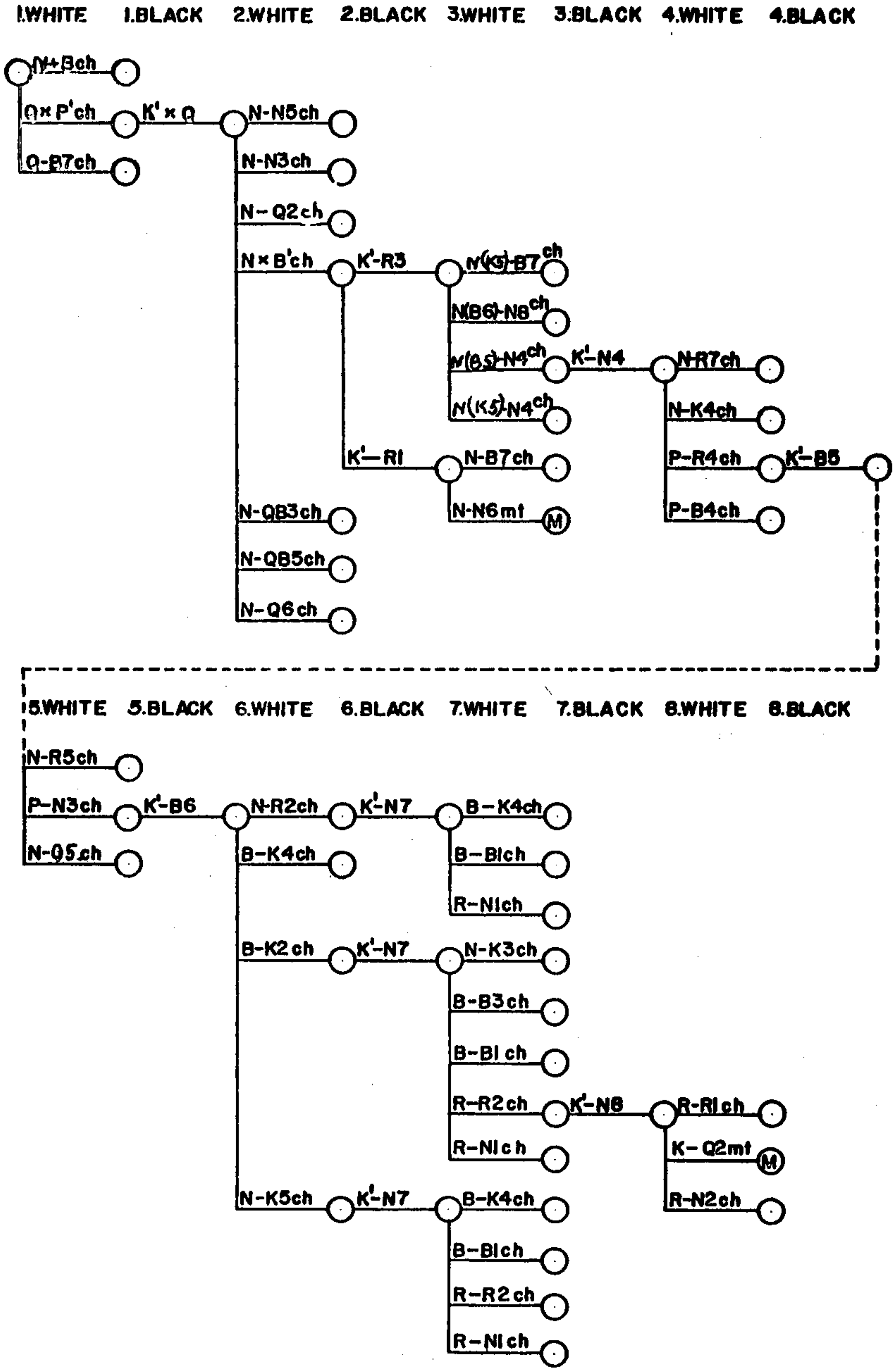

Figure 2: Exploration tree of mating combination program

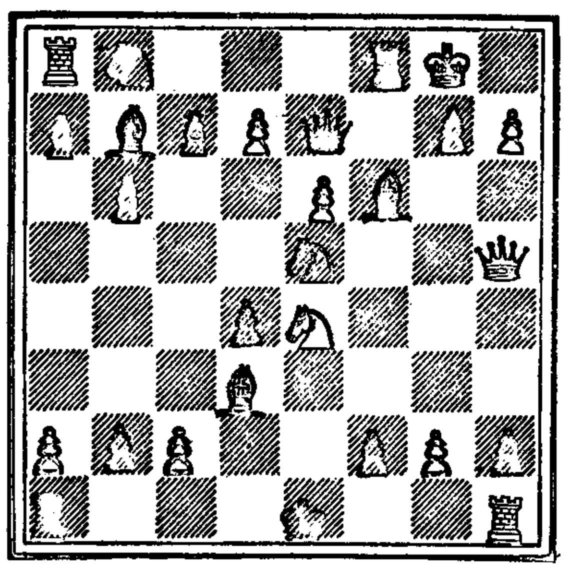

Figure 2 (continued)

Thomas (Simon and Simon, 1962). The program, in this instance, discovers a combination that is eight moves deep after examining only 52 branches of a search tree. The heuristic chiefly responsible for this program's power is the rule that it explore first those moves to which the opponent has the fewest replies. This heuristic is derived from the idea that only if the opponent is highly constrained will a mate be possible; it has the important side effect that it limits the tree to an almost constant width.

Chess Playing by Humans

Few studies have been made of the psychology of playing chess. The most important work is that of de Groot (de Groot, 1964); most of the other studies are so general in their comments as not to be relevant to our discussion. De Groot examined in great detail the protocols of a number of chess players, who were analysing positions in order to choose a move. Their skill ranged from good club players up to grandmasters. The position we will analyse here was taken from de Groot's book (Position A), and perhaps the chief difference between our procedure and his is that we used a tape recorder, whereas de Groot did his work before such devices were readily available.

The general characteristics of our subject's protocol agree well with de Groot's analysis, and a summary of de Groot's main points will serve to summarize the gross features of our subject's behavior as well.

Selective Search. Humans playing chess spend a very substantial amount of their time searching for the consequences of the moves they are considering. The search is highly selective, looking at only a few of the multitude of possible continuations. There is no evidence that the total number of different positions considered by a player, during an analysis lasting up to fifteen minutes or so, exceeds one hundred. Evidence as to how many positions are considered is obtained by protocol analysis of the sort we will conduct below. The estimates err on the low side, since players may fail to mention positions they consider, but by no stretch of the uncertainties could the estimates be more than doubled.

Both facts mentioned above -- that players do search, but that they search only a small space -- are important in assessing existing chess programs as descriptions of human chess playing. Programs that search thousands or tens of thousands of positions per analysis are almost certainly proceeding quite differently from humans. On the other hand, those programs that search much more selectively may be more relevant.

Elementary concepts. Superficially, the same kinds of elementary chess concepts are involved in human play as appear in chess programs: attacks, defenses, pins, open files, isolated pawns, and so on. This is to be expected, of course, since the chess programs are written, by and large, by chess playing programmers and rely heavily on the standard chess literature. However, the occurrence of the same concepts in both programs and protocols does not imply that they have precisely the same extension, nor (more important) that the concepts are used in the same way in both analyses.

Global concepts. There are concepts in human chess playing that are much more global than those above; for example, "a developed position," "control of the center," "a won position," "a weak King side," "a closed position." Their counterparts in current chess programs occur mostly in the static evaluation processes, and no one maintains that the correspondence is very close. To date the work on chess programs has not shed much new light on these higher-level concepts. More generally, psychology has had little to say about how global concepts organize behavior.

Episodes. Human chess analysis is broken up into separate episodes. As one would expect in a task in which the subject is plunged into a complex situation, initially he orients himself to the board position. He also sums up at the end in deciding on the single move he will play. The number of episodes between these boundaries is variable and depends on the level of analysis. De Groot distinguished three major episodes, which he called phases (he included the initial orientation as a fourth): exploration, elaboration and proof. But within these phases many discontinuities in the problem solving process occur which mark the boundaries of still smaller episodes. More important are the characteristics of different types of episodes, discussed below.

Progressive deepening. There often occurs what de Groot calls progressive deepening: the analysis of a move is reworked repeatedly, going over old ground more carefully, exploring new side branches and extending the search deeper. Indeed, some players start by conducting a sample variation to orient themselves to the position. This idea of "rough cut, fine cut," to use a term proposed by J. C. Shaw, is not prominent in chess programs, although there have been some proposals along this line. The MIT program referred to earlier does examine all legal moves statically to select out the plausible moves to be considered to greater depth.

Exploration and verification. Some episodes are devoted to exploring for new information; others are devoted to proving or disproving a hypothesis (i.e., to verification). Search may be conducted in quite different ways in the two cases, since the information sought is different. The general tendency in human behavior to deal with a complex world by a sequence of singular hypotheses rather than by narrowing possibilities deductively using the full amount of incoming information, is well attested (Bruner, Goodnow and Austin, 1956). In general, the human player has no way to squeeze all the information out of each new observation on the board; dealing with hypotheses seriatim throws away much information but makes the cognitive task manageable. Although there are programs that create and test hypotheses (Feldman, 1963; Kochen, 1961), existing chess programs make no use of these mechanisms.

Problem definition. Human chess players periodically attempt to redefine what the problem is. The redefinition is usually a conclusion based on the immediately prior analysis and is accepted as the new working assumption. These summaries are put forward in rather general terms e.g., "In any case White will have to extract some profit from that weakness after all." One might be tempted to think of this as hypothesis formation, but it is not. Hypotheses can arise (and be accepted and hence worked on) without deliberate summarization; and likewise, after an attempt to redefine the problem, there is normally no testing activity devoted to proving or disproving the efficacy of the new problem definition. Nothing of the process of redefinition occurs in current chess programs.

Position evaluation. A major difference between human play and most chess programs lies in the evaluations of positions. The evaluation by de Groot's subject's were often rather elementary, mentioning a single advantage (e.g., "and Black gains an open file"). This is in contrast -- although not in contradiction -- to the rather elaborate polynomial evaluations that have been used in most chess and checkers programs. In human play there seldom occurs a balancing of many factors, some pro, some con, to arrive at an over all estimate. Sometimes de Groot's subject used very global phrases such as "... and it's a won position for White," where it is not possible to see what structure or feature of the position leads to the evaluation. However, human players generally make evaluations at the terminal positions of each line of search (the static positions of chess programs) and make no evaluations at intermediate positions (the dynamic position of chess programs). In this respect, players and programs agree.

Perceptual processes. All of the features we have mentioned will be illustrated by our subject's protocol. One other feature of human play, discussed in detail by de Groot, will be absent. In trying to find measures to distinguish strong from weak players (other than making the correct move), de Groot was singularly unsuccessful with the statistics of search and analysis -- e.g., the number of positions examined. (However, the worst of de Groot's players were good enough to play occasionally in local tournaments in Amsterdam.) He finally succeeded in separating strong from weak players by using perceptual tests involving the reproduction of chess positions after brief exposure to them (3-7 seconds). The grandmaster was able to reproduce the positions perfectly, and performance degraded appreciably with decrease in chess ability. De Groot was led to propose that perceptual abilities and organization were an important factor in very good play. Since the protocal examined here provided no direct evidence on perceptual processes, we will be silent about this possibly important aspect of human play.

With the picture just presented, one might feel that there is little in the way of correspondence between computer program and human chess playing, beyond the fact that both search a good deal and use heuristics, based on the same elementary chess concepts, for controlling that search. However, the real issue is not one of exact correspondence. Rather, the question is whether our current knowledge of information processing, as expressed in problem solving programs, will let us shed some light on how humans play. The information presented so far only serves to highlight certain gross aspects of human chess playing behavior. A much more intimate view is necessary before most of our knowledge can be brought to bear. To this we now turn.

A Subject and a Position

Figure 3 shows a middle game position taken from de Groot. The subject is confronted with it and asked to choose White's next move. He is allowed to take as long as he wants, which in practice means "a thorough analysis

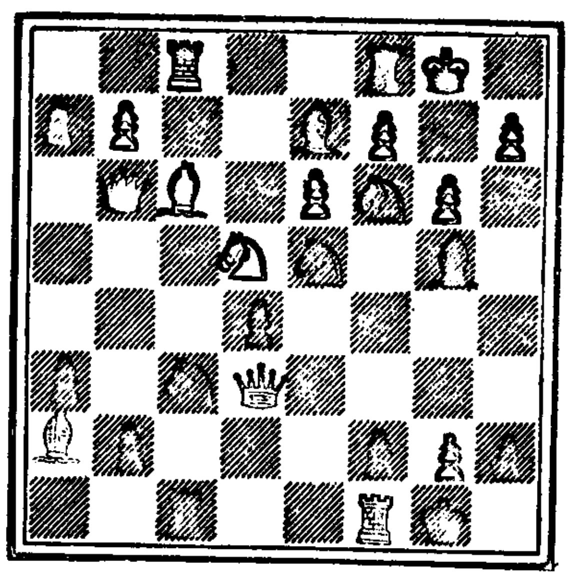

Figure 3: Position A (from de Groot)

within the limits of normal over-the-board play." The subject is asked to talk aloud as he makes his analysis; his words are recorded on a standard tape recorder.

The subject is a chess player of moderate caliber, who was active in a college chess club at the time the protocol discussed in this paper was recorded. He was by no means an expert or Class A player, as these terms are used in chess ratings, but his play is undoubtedly much better than that produced by existing chess programs.

The Position. As an introduction to the position of Figure 3 we can do no better than quote de Groot's opening comments upon it:

> ...Taken from a game between A. D. de Groot - C. Scholtens, April 10, 1936. White is on move. ... This position mainly presents problems of a tactical nature. Through his last move (...Q-N3) Black has created a "hanging position" for his Bishop on K2; it is defended only by the exchangeable Knight on Q4 so that the Black Knight on B3 is somewhat tied down. There are all sorts of exchange possibilities in the center and the question is whether or not it is possible for White to make some profitable use of the tactical weaknesses in Black's position. If no such possibility should exist, White could best strengthen his position with some calm move.

> From a thorough analysis, however, it appears that White is in a position to get the better of it; there is even a forced win. The winning move is 1.BxN/5. ... (de Groot, 1964, sect. 26).

The move 1.BxN/5 was chosen by four out of five of the grandmasters who analysed it for de Groot. It will be noted that this is the move selected by the subject. However, it cannot be concluded from this that his analysis is correct in its details; in fact, the subject appears to remain ignorant of several of the essential features of the position.

The Protocol. The appendix gives the transcription of the subject's protocol. His words lie along the right half of the page and an encoding of the content of his remarks lies only the left half. This encoding is accomplished manually according to a scheme to be described in another place[4]. It serves here merely as a convenient condensation of the chess content of his remarks. It is largely self-explanatory; the few notations that are necessary to its understanding being given on the first page of the appendix. Each line of the code contains a single elementary assertion, considered action, or self-asked question. The fineness of division is related both to putting into separate lines those comments that could reasonably be said to have occurred in sequence rather than simultaneously, and to naming each element of the entire protocol.

[4] The coding use in this paper is a variant of the actual scheme, in order to facilitate reading without extensive knowledge of the coding language.

The ease and reliability of coding varies considerably. Easy to code are phrases such as B16, "and the Bishop at Rock 2 is bearing down on the Knight," which is coded as "B(R2) bear on N'." As in standard chess notation, the N' is completely identified by the board context. It is necessary to define the relation "bear on" in a rigorous way, but there is excellent consensus in the chess world on the term, and a check of all occurrences in the protocol reveals no idiosyncratic use. Hard to code are utterances such as B138-B142 "Ah, let's see, we will play Knight takes Knight - play Bishop takes Knight. Bishop takes Knight - Knight takes Bishop. Then where do we stand - then we play Knight takes Knight and Black will play Pawn takes Knight." The question is, which moves are corrections of previous moves and which are subsequent moves. The interpretation shown in the code happens to be the only one consistent with the entire context, but real ambiguity can occur. A different kind of difficulty is shown in B134, "Now, Black's Kingside is in sad shape." It happens that the statement is not objectively true for most reasonable definitions of "sad shape." The subject does expand in B135 (which is precise) but the question remains whether B134 is just a prestatement of B135 or whether a more general concept is intended. Since no other occurrences of the phrase (or highly similar ones) exist in the protocol we are left at sea. Despite such difficulties, most of the protocol is readily coded.

Turning to the grossest features of the subject's behavior, we notice that he worked on the problem for almost 17 minutes, and that there appeared to be no difficulty in inducing him to talk. His average production of words is 115 per minute, ranging from a low of 90 to a high of 145. Chess has a well developed argot for describing positions and their analyses. Thus the subject, who is a fluent in this chess language, produces a stream of talk that is completely task oriented, and singularly free from stumbles, breaks, and frustrated attempts at expression. All of the designatory phrases used in chess ("Bishop at Rook 2," "Knight under single attack," "double up rooks on the Queen Bishop file ") are immediately at his service[5].

Perceived relations and dynamic analysis. The protocol of the subject mentions both moves and relationships on the board, the latter, of course, ultimately deriving from moves. Thus, B22 says "The Bishop at Rook 2 can take the Knight"; and from B23, "which would no doubt be answered by ...," it is clear that the making of a move is being considered. On the other hand, B5 says "his Queen is threatening my Knight's Pawn." This is true because Q'xNP is possible, but it does not mean that the move has been considered.

This distinction between moves and perceived relations is based, not on features of the board, but on characteristics of the information system that is processing the board. Like a system of axioms, all the future implications from any exploration in the game tree are "contained" in the present position (indeed, consideration of move sequences is just a way of extracting these remote relations). Any of these implications could be "statisized" and made

[5] However, the total rate of flow of about two words per second is also attained by subjects in tasks where they have much more difficulty expressing themselves, providing one counts all the words, independently of whether they are used in complete phrases or not.

an object of present perception. Thus, "potential forking square" is just a description in terms of the present position of a situation that could be realized by several moves (i.e., by moving an appropriate piece to the square in question). Likewise, one could think of relations such as "the ultimate checkmater of the King," as being as much presently perceived as "the attacker of the Rook." That we normally know of no way to discover the former relation except by searching the tree of moves is only a limitation on ourselves (equivalent to the limitation on the beginner in discovering forks). Indeed, there are places in the endgame where a piece can be assigned the property of the "ultimately promotable Pawn," without examination of the forward move tree in the sense of a search.

We have labored this point at length, since our analysis of the subject's search behavior will depend critically on our distinguishing where he preceives a relation and where he considers a move. Our encoding of the protocol makes a choice in each case between these interpretations. ("PxN" and "Q-QB3" being examples of considered moves; "attack" and "pin," being examples of relations). In practice making the distinction is not difficult for this protocol.

The Drama. Before taking up the analysis of the subject's behavior it is first desirable to obtain an overview of it. In agreement with de Groot's findings, our subject's behavior can be divided into a series of episodes, which we have labeled E1 to E25. Although presuming the analysis yet to come, these episodes permit us to give a meaningful picture of what went on during the entire problem.

This description, given below, is appropriately enough viewed as a drama in which the subject struggles to discover which are the good and which the dangerous things in the situation. Since twenty-five episodes are too many for the reader to keep in mind (the subject did not have to keep them in mind, he only had to live them), we have grouped these into seven still larger scenes. These scenes do not correspond to problem solving phases or stages; rather each is simply a set of explorations that are under control of a common aim. Occasionally, however, interruptions can occur during a scene which are not devoted to the main concern of a scene.

Scene 1: Orientation (0' 0")
"OK, White to move... in material the positions are even."

E1: Examines first the material situation, then (systematically) enumerates Black threats, then White ones. Is aware of Q'xNP threat.

Scene 2: Explore 1.BxN/5 (1'20")
"The Bishop at Rook 2 can take the Knight, which would no doubt be answered by ..."

E2: Traces exchange until Q' driven back to defend against double attack of Q attack P' and Q and B attack N'.
E3: (Interrupt) Explore to see if Q-B3 (discovered in E2) is a good initial move; answer is negative.
E4: Retrace exchange, re-examining arguments for Black's choices; conclude White wins a P.
E5: Retrace exchange, examining counterattacks (3...Q'xNP and 3...Q'xQP) after 3.Q-B3; conclude White wins a piece for a P in this case.

Scene 3: Search widely (5'0")
Let's see if there's anything else here."

E6: Explore 1.NxB': nothing.
E7: Explore 1.NxBP': nothing.
E8: Explore 1.NxNP': nothing.
E9: Explore doubling Rooks on QB-file: nothing.
E10: Explore K'side attack with Pawns: nothing.

Scene 4: Re-examine 1.BxN'/5 (6'25")
"the immediate exchange seems indicated if we can win a piece for a Pawn."

E11: Retrace exchange, examining immediate counterattack (1...Q'xNP); White wins a piece for a Pawn or two.
E12: Retrace E11, examining a possible pin against White; conclude there is no threat.
E13: Retrace exchange, considering retake by N'(B3), which apparently leaves B'(K2) undefended; discover N' still defends B'(K2) (from Q4), but sees how to continue exchange and keep own B(N5) unthreatened.
E14: Retrace original variation, but consider recapture by N'(B3) (discovered possible in E13) later in exchange; conclude the whole 1.BxN'/5 exchange is worth nothing for White.

Scene 5: Try something else (9'0")
"Now, Black's Kingside is in sad shape -"

E15: Discover mating configuration (B-R6, Q-N7); B is well placed, but not easy to get Q in place; conclude that B'(K2) is difficulty.
E16: Explore 2.N-K4 in an attempt to get rid of B'(K2); conclude move is fruitless.
E17: (Interrupt in middle of E16) Examine whether 1.N-K4, which reveals R bear on B(B3), imposes a pin on Q' so it cannot capture NP; conclude no pin (return to E16).
E18: Worry about 1...Q'xNP after 1.N-K4, which threatens B(R2); see that B must move and BxN' only reasonable alternative; conclude that 1.BxN/5 should be initial move.

Scene 6: Return to BxN'/5 (12'15")
"...so let's take the Knight right away."

E19: Review responses to 1.BxN'/5; conclude that all lead to complications (which summarizes past explorations).
E20: Examine 1.BxN'/5, B'xB; try new alternative for White (2.N-R4).
E21: Retrace E20; conclude advantage is with White; Black will not respond 1...B'xB.
E22: Examine 1.BxN'/5, P'xB; conclude the advantage (isolated P') is with White, so Black will not respond 1...P'xB.
E23: Examine 1.BxN'/5, N'xB; conclude that 2.NxN' makes this impossible for Black; hence Black will not respond 1...NxN.
E24: Conclude from E20-E23 that Black must play 1...P'xB; explore gain in terms of K'side attack.

Scene 7: Decide on BxN' (16'20")
"...so the best move is then Bishop takes Knight."

E25: Make decision and give next move, conditional on Black's response.

Search Behavior

If we put together all the moves that the subject considered, we obtain the tree of exploration shown in Figure 4. This tree might have been badly discontinuous, with connecting branches missing due to silence on the part of the subject while traversing them. In actual fact, all the nodes in Figure 4 are mentioned explicitly by the subject, with the exception of the four enclosed by < >, which are inferred. The dotted lines indicate cases where it is inferred that the subject did not propose a move; likewise, where non-specific moves are given -- e.g., Q-move -- it is inferred that the subject was no more specific than is stated. The tree contains 64 positions, including the current one. There are also eight moves which are distinctly generated, but where it is inferred that the positions from those moves are never considered; these are indicated by branches with no small circle at their tips (e.g., R-exchange at the top of the Figure). The number of positions considered is well within the figure of 100 quoted earlier as an empirical upper bound to human over-the-board search, and is roughly comparable to the numbers of positions considered in Figures 1 and 2. Thus, the subject examines about four new positions per minute.

Episodes and progressive deepening. The tree of Figure 4 does not reveal the way in which the tree is generated. In Figure 5 we have depicted

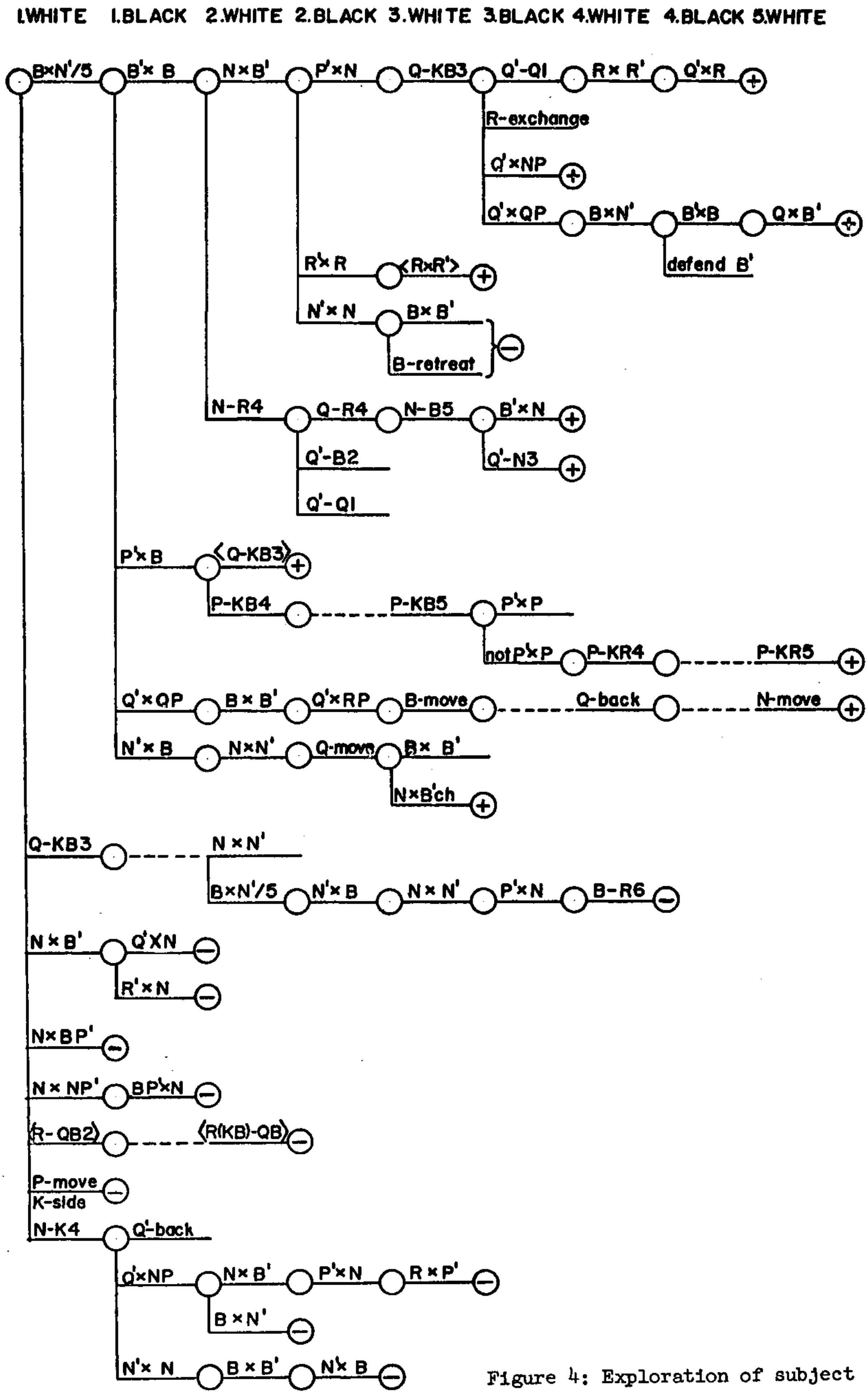

Figure 4: Exploration of subject

the search of our subject in the time order in which it is performed. What is higher in the Figure takes place first, and as before, the search takes place from left to right. Thus the moves considered begin (E2) 1.BxN'/5, 1...P'xB, 1...B'xB, 2.NxB', 2...R'xR, 3.< RxR' >, 2...P'xN, 3.Q-KB3, 3...Q'-Q1, (E3) 1.Q-KB3, etc.

We see immediately a distinctive pattern. The subject searches extremely deep without any appreciable branching (mostly without any branching at all). At the termination of the search, he returns to the current position and starts over. Often he reconsiders an initial move already analysed (among the 23 starts there are only 8 distinct initial moves). Only when we get to E18 and beyond does the subject not go back to the start, but instead picks up at the point of the opponent's first response. Three of these cases (E20, E24, E25) have the same initial move (BxN'/5) which has already occurred eight times; the other (E18) begins with N-K4, which has just occurred twice. As with the tree in Figure 4, all of the moves are explicitly mentioned, except those in < >, so the evidence for returning to the start is direct. Thus, in search E24 there is no evidence for any specific consideration of the opponent's replies; only of the subject's own positive moves (B224 to B229).

The almost uniform return of the subject to the base position after each burst of exploration offers the means of segmenting the total problem into twenty-five episodes. E1, the orientation phase, is not shown in Figure 5. The boundaries of these episodes are marked, not only by the discontinuity in the position considered, but by evaluative and summarizing statements terminating an episode, and by proposals about what is to be done

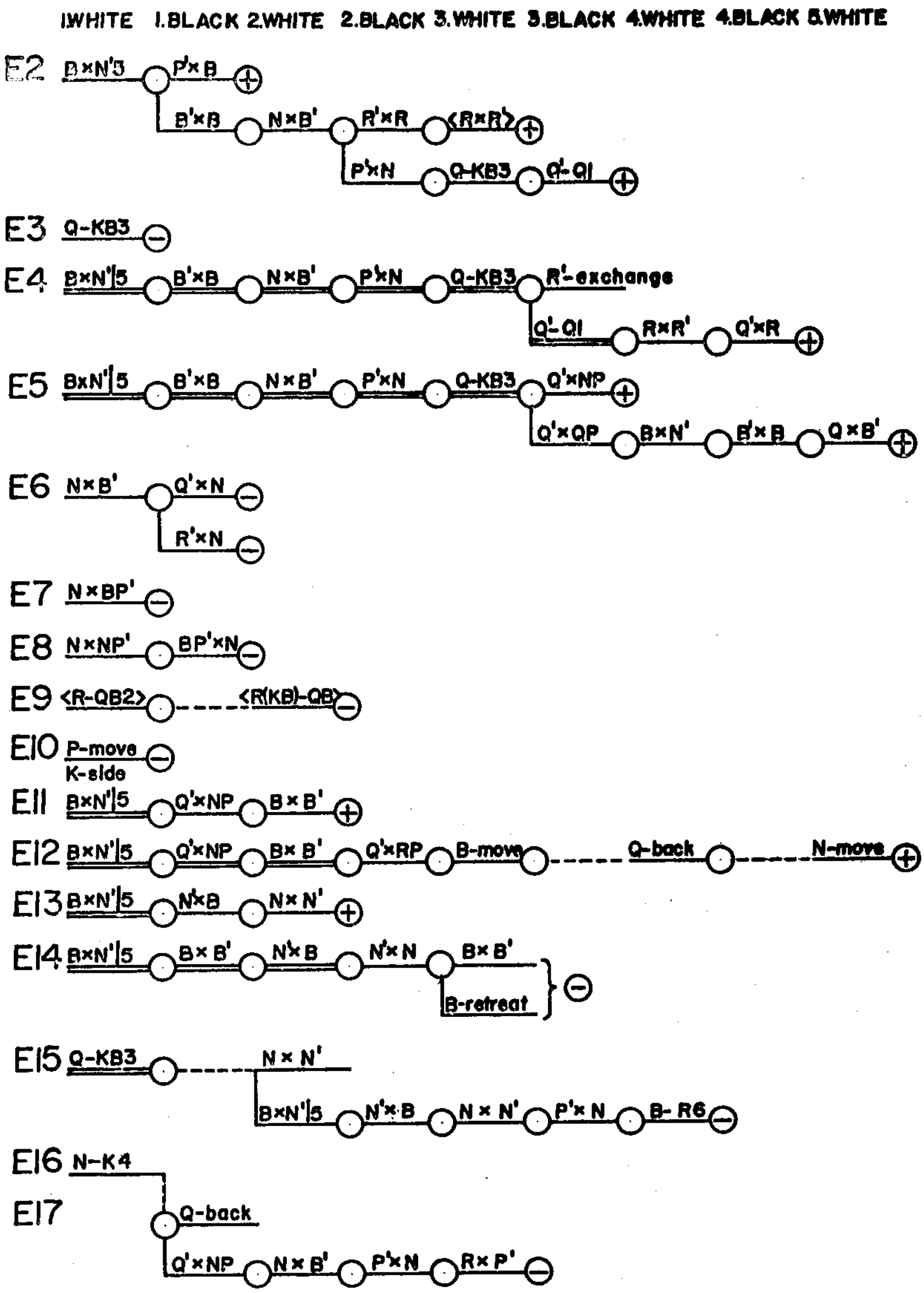

Figure 5: Explorations of subject by episode

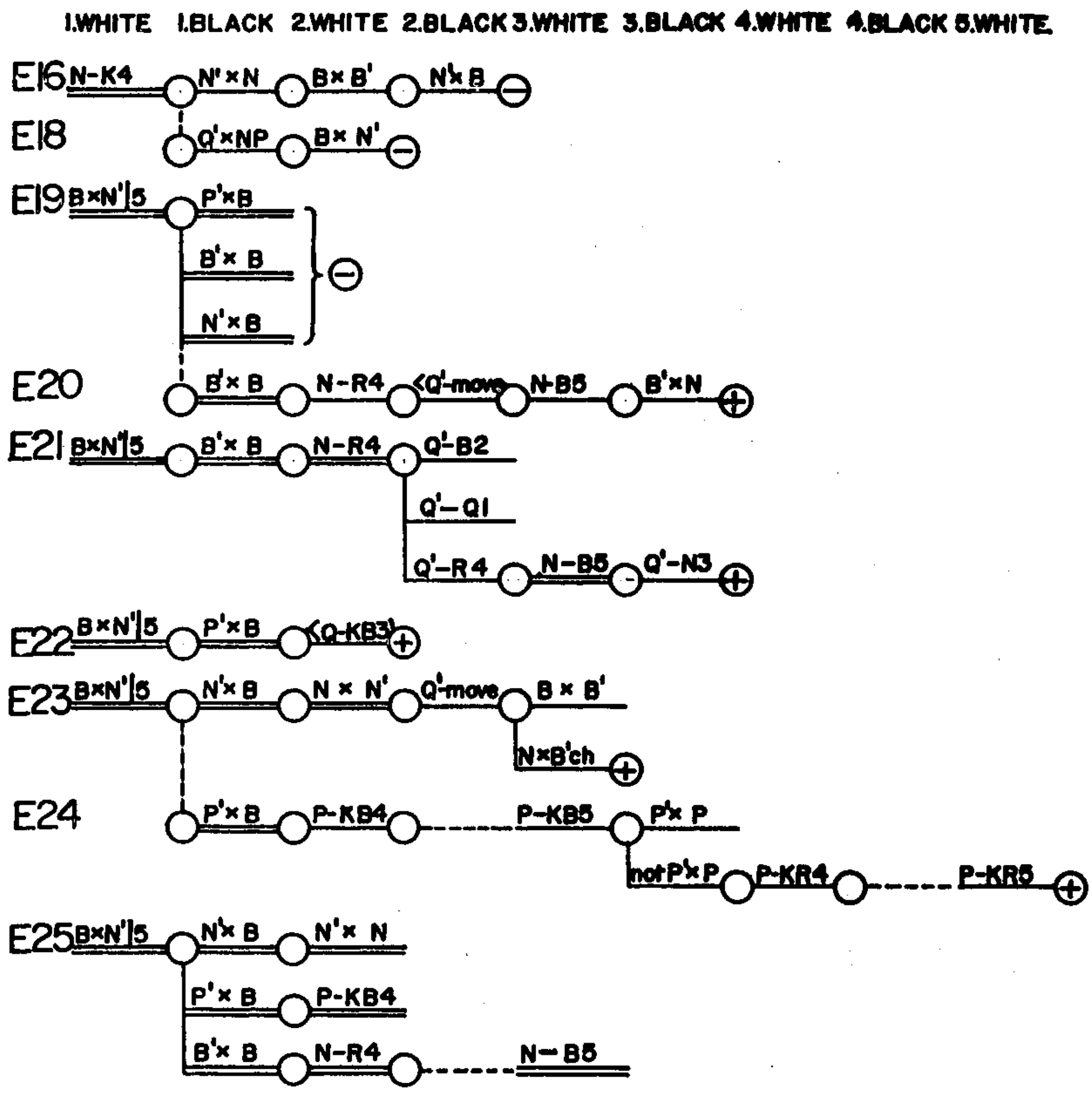

Figure 5: Explorations of subject by episode
(continued)

next initiating an episode. Thus there is little doubt about the reality of the episodes in the organization of the subject's behavior.

The behavior shown in Figure 5 fits well what de Groot called progressive deepening and broadening of the investigation. Each reworking of a path starting with the same move may be viewed as an attempt to deepen the state of knowledge about that possible choice. Some secondary searches penetrate deeper in terms of number of moves; others search off in new directions. We have marked with double lines those moves which are retracings of previous moves, so that one can see clearly where the exploration takes a new turn.

Search strategies of programs. We can now ask what sort of an information processing organization could have produced the search behavior shown in Figures 4 and 5. Whereas in describing the chess programs we could analyse the internal structure of existing programs and list the main mechanisms we discovered, here we must hypothesize mechanisms and then ask what behavior it leads to. At best, we can show that our hypothesized organization is sufficient to reproduce the subject's observed behavior. We cannot show it is necessary, although we may be able to show that some alternative organizations are incompatible with the behavior.

In designing search programs it is useful to distinguish the strategy of search from the information that is gathered during the search. The search strategy tells where to go next, and what information must be kept so that the search can be carried out. It does not tell what other information to obtain while at the various positions, nor what to do with the information after it is obtained. There may be strong interaction between the search itself and the information found, as in the decision to stop searching, but we can often view this as occurring within the confines of a fixed search strategy.

In the description given earlier of chess programs, the search strategy was left implicit. In fact, there is some freedom of choice about how to put together into a complete program all the pieces mentioned there. The most common strategy is the Depth-First strategy. We may describe it by the following schema:

Depth-First Search Strategy

In considering a position, X:

- All positions that led to X are available.
- If X is static, then return to the position that immediately led to X.
- If X is dynamic, generate all the moves to be considered from X. Consider each of the positions from these moves in turn.
- When through, return to the position that immediately led to X.

In Figure 6 we show by the arrows the path the Depth-First strategy would take in generating an illustrative tree. (Note that depth in the tree runs from left to right in the figure.) Once a particular position has been generated, all deeper search beyond the position is carried out before that part of the tree is abandoned. This procedure is highly efficient memorywise, in that only a single line of positions from the base position (i.e., the one actually on the board) up to the position being considered needs to be kept in memory at a given time. (Usually, of course, only the moves are kept in memory, not the full positions, since the prior positions are regenerated from the current

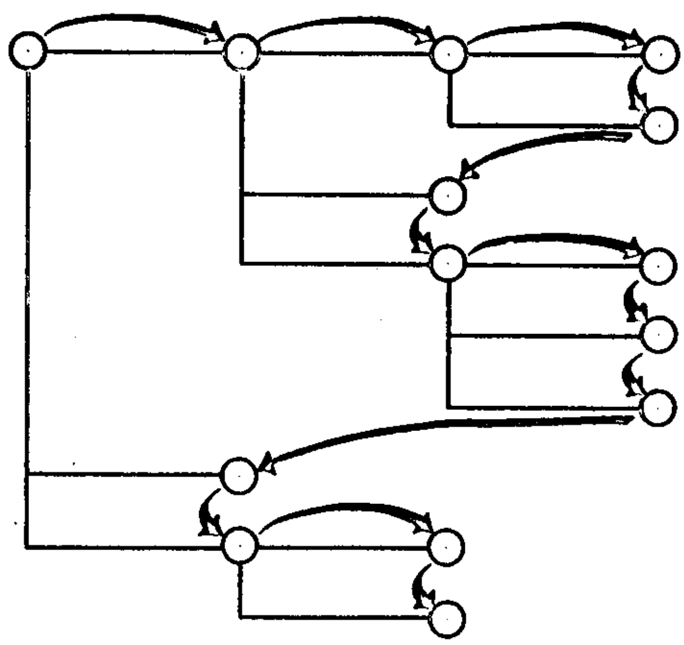

Figure 6: Depth-First search strategy

position and the move that led to it.) The Depth-First strategy is particularly suited to the requirements of the minimax inference procedure. Minimaxing derives the value of a dynamic position from the values of all the positions one level deeper that are considered. The Depth-First strategy is exactly that strategy which makes all these values available at one time with a minimum of memory and retrieval effort.

For comparison, consider an alternative strategy, given by the following schema:

Breadth-First Search Strategy

In considering a position, X:

- All positions generated before X is considered are available.
- Generate all moves to be considered from X and store all the positions from these moves.
- When through, consider a stored position that is at the same depth as X. If none exist, consider a stored position at the next level deeper.

In Figure 7 we show the path of the Breadth-First strategy in generating the same illustrative tree used in Figure 6. Instead of going deeper and deeper, it completes all positions at one level before going on to the next. To do this, of course, all positions must be stored until they are considered. In compensation, the Breadth-First strategy avoids looking too deep in one part of the tree when something obvious is awaiting discovery at level 1 or 2 in an unexamined part of the tree. The Breadth-First strategy has been used in some theorem proving programs, but has not been used in any game playing programs.

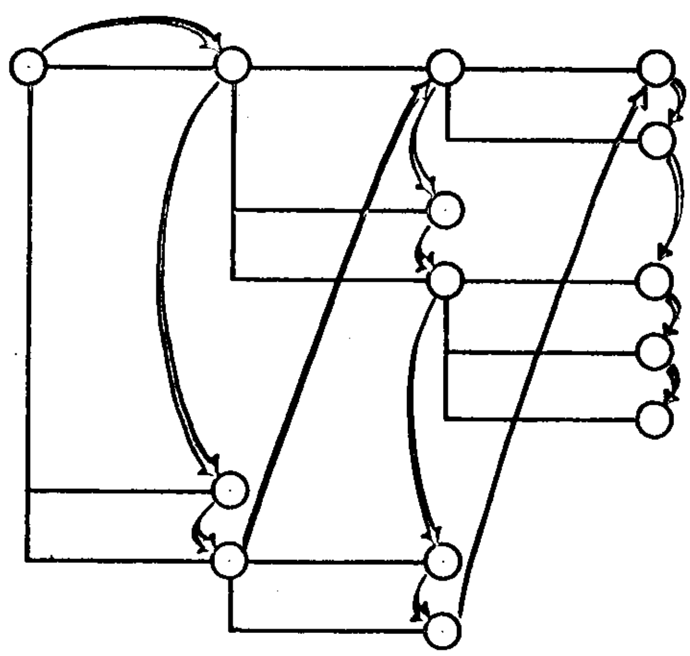

Figure 7: Breadth-First search strategy

Minimaxing with this search strategy, even though it can be done, requires more memory, effort and organization, than with the Depth-First strategy.

Search Strategies of Subject. This excursion into the search strategies of programs is intended to lay the groundwork for considering what search strategy might generate the trees of Figures 4 and 5. Clearly neither Depth-First nor Breadth-First will do. As a start, consider the following strategy:

Progressive-Deepening Search Strategy

In considering the base position:

- Either an old base move (i.e., one from the base position) is reconsidered, or a single new base move is generated and considered.
- A summary is kept of the state of analysis of each base move.

In considering a position, X, other than the base position:

- The state of the analysis of the base move leading to X is available.
- If X is static, return to the base position.
- If X is dynamic, generate a single move from X and consider it.

Given this strategy, we would expect sequences of linear searches without any branching at all, going as deeply as necessary to get information. The summaries of the current state of analysis of base moves permit different moves to be generated upon successive visits to the same position. The Progressive-Deepening strategy by no means describes Figure 5 exactly, but it is surely a closer approximation than either Depth-First or Breadth-First. One virtue of the Progressive-Deepening strategy lies in lowering what Bruner called "cognitive strain" (Bruner, Goodnow and Austin, 1956). No complicated internal housekeeping is needed to keep track of where the search is. Only a single position need be stored internally, and it can be stored in terms of the way it differs from the base position, which is under continual surveillance (thus providing a continuing memory of all the things that have not changed). The search strategy is even "fail safe" in that if something goes wrong with an exploration -- e.g., the subject loses track of what the current position is -- then the search is simply terminated and the total analysis continues. The only loss is the effort spent in the abortive exploration.

To approach one step closer to what the subject was doing, let us define another schema:

Modified Progressive-Deepening Search Strategy

In considering the base position:

- Handle the same way as the Progressive-Deepening strategy.

In considering a position, X, other than the base position:

- The state of the analysis of the base move leading to X is available.
- If X is static, return to the base position.
- If X is dynamic, generate a set of moves from X. Consider each in turn, but only statically. Select one of these moves and consider it dynamically.

This strategy will produce a sequence of explorations, each starting from the base position, but with the tree of each exploration resembling a skinny Christmas tree: the tree would have a main trunk, and at each node there would be a tuft containing a number of branches, each one move deep.

If we examine Figure 5 we see that of the 15 instances of multiple branching 11 are exactly of this form. The contrary examples are in E2, E18, E24, and E25. Of these, E2 is ambiguous. The subject referred to an exchange of rooks (B28), and we are left to infer either that he considered first 2...R'xR and then 3.RxR', or that the action of exchanging was a single conceptual act in some sense. We have already commented on E18, and E24, which appear to be truly cases of dropping back to the position defined by a familiar move. A similar comment applies to E25, which is a summary of the subject's behavior under different contingencies, and again drops back each time to the position after 1.BxN'/5. Thus, in terms of two features of the search -- return to base position and single level tufts at each node -- the Modified Progressive-Deepening (MD) strategy seems a plausible description.

Given the search strategy, a number of additional processes must be specified in order to make a complete chess playing system. Some of these fill out the parts specified in the strategy; others determine what additional information is to be gathered and to what use it is to be put. These parts form a single system, so that the shape of each one depends on the others. Thus our order of analysis in the paper begs somewhat the question of which features determine the others -- of which are chickens and which eggs.

Episode Generation

According to the MPD strategy, episode generation and move generation within an episode are interwoven, since the moves generated are determined by both the current position and the present state of the base move that is being explored. We first examine the episodes as wholes, viewing them as providing the context within which specific move generation operates.

Functions of episodes. Since the subject makes a number of explicit statements about the function of each episode, the nature of an episode does not have to be inferred completely from the pattern of moves shown in Figure 5. The comments, which have already been reflected to some extent in the recitation of the drama, not only makes good chess sense and good problem solving sense, but are consistent with the subject's behavior throughout the episode. Consequently, we can accept the naive hypothesis that the episodes function in the total problem solving attempt pretty much as the subject indicates. (The whole set of protocol statements is sufficiently interdependent, that it does not appear easy to manufacture radically different interpretations for the episodes.) Figure 8 provides for each episode a statement of its function, and its outcome (as positive or negative for White). We have also noted additional information that was discovered where this is relevant to later episodes -- e.g., that Q-KB3 was discovered during E2. In the figure, the term "explore" means to go down a new path; and the term "rework" means to go down the specified path again, for whatever reason. For the Rework episodes we have added a brief characterization of what happened during the episode.

Rules for episode sequence. To some extent each of the episodes is unique, especially when put in the context of the previously occurring episodes. Nevertheless, it is possible to write down some rules that would generate a sequence of episodes not unlike that shown in Figure 8. These rules are concerned with which base move is selected for the next episode and what context governs the exploration of subsequent episodes of the same base move. The rules do not describe behavior within an episode. Six rules are given below, and the protocol will be examined to see whether these rules are reflected in the subject's behavior. There is substantial evidence that the subject is observing the first three rules; and more limited evidence for the remaining three.

R1: The analysis of each base move is independent of the analysis of other base moves, except that it can be interrupted by other activity. That is, each episode in the analysis of a base move is determined only by the results of the prior episodes of that base move.

R2: The first episode of a base move employs normal moves, and subsequent episodes utilize increasingly unusual moves. ("Normal" and "unusual" will be discussed below).

R3: If the evaluation of an episode gives a favorable result, the analysis of its base move is continued; if the evaluation is unfavorable, a different base move is analysed.

R4: When exploring, moves for the opponent may be considered that are favorable to self (in order to place an upper bound on the possibilities).

R5: The analysis of a base move will be interrupted to pursue other moves, discovered during the episode, that seem to have merit either for self or for the opponent.

R6: Before a base move is finally chosen, a check is made for other alternative base moves.

Verification of rules. Figure 8 shows, for each episode, which rules are exemplified by that episode, and whether the episode is confirming or disconfirming of the rule. There is not space (nor reader's patience) to deal individually with each of the 6x25 judgments, but let us note in general how each of the rules relates to the behavior.

R1 asserts that the total analysis can be factored into a set of little analyses, one for each base move. The interaction between them is only one of

Episode	Function	Result	R1	R2	R3	R4	R5	R6
E2	Explore 1.BxN'/5	+ new move Q-B3		+				
E3	Explore 1.Q-KB3	-			p		+	
E4	Rework E2 (extend)	+	+	+	+			
E5	Rework E2 (counterattack)	+	+	+	+			
E6	Explore 1.NxB'	-		+	p			+
E7	Explore 1.NxBP'	-			+			
E8	Explore 1.NxNP'	-		+	+			
E9	Explore Rooks on QB-file	-			+	+		
E10	Explore K-side P-move	-			+			
E11	Rework E2 (counterattack with loss)	+	+	+	+			
E12	Rework E11 (extend)	+	+	+	+			
E13	Rework E2 (recapture with loss)	+ new move N' recapture	+	+	+			
E14	Rework (N'recapture)	-	p		p		+	
E15	Explore Get B-R6, Q-N7	- need remove B'(R2)		+	+	+		
E16	Explore Remove B'(R2)	discover attack on B'(B3)						
E17	Explore Attack B'(B3)	-		+			+	
E16	Explore 1.N-K4 (continued)	-		+	+	+		
E18	Rework E16 (counterattack)	- BxN'/5 necessary	+	+	-	+		
E19	Summarize 1.BxN'/5				+		+	
E20	Rework E19(B'xB)(explore 2.N-R4)	+	+	+	-			
E21	Rework E20 (extend)	+ not 1...B'xB	+	+	+			
E22	Rework E19(P'xB) (extend)	+ not 1...P'xB	+	-	+			
E23	Rework E19(N'xB) (extend E13)	+ not 1...N'xB	+	+	+			
E24	Rework E22 (extend)	+	+		+	+		
E25	Choose 1.BxN'/5							-

Legend for rules: + confirm
- disconfirm
p another rule has priority

Figure 8. Functions of Episodes

allocation of effort, including the decision to abandon the analysis of a base move because others have proved better.

> It is difficult to refute this rule; one would have to find features within an episode derived from sources other than the prior episodes of the same base move. Presumably one could recognize them if they occurred, but it is not easy to imagine examples. One apparent exception to R1 comes from the transposition of the move, N' recaptures, from E13, where it was discovered, to E14. However, we view this instance as showing the priority of rule R5 (interrupting). It could hardly be considered a counter example to R1 since E13 is part of the analysis of E2. Setting levels of aspiration (used in evaluating each episode) on the outcome of the episodes of all base moves might be considered counter evidence to Rule 1, but discussion of this point will have to wait until the section on Evaluation.

R2 specifies the dependence of an episode on prior episodes of the same base move. It asserts that exploration goes from the "normal" to the "unusual." The underlying model will be elaborated in the section on Move Generation; only the gross outlines are needed here. Consider the following responses to an attack: defend; counterattack threatening equivalent material; counterattack threatening less material; move and ignore the attack. The "normal" response to an attack is to defend the man attacked. Each of the other responses is more unusual, and increasingly so, although they may be the correct response in the situation. R2 asserts that the subject has such a model of normal and unusual responses, and that each successive episode of the same base moves involves considering more unusual responses. The rule considers a move to be unusual if it is made again after prior analysis has shown the continuation to be bad for the side making the move. This shows up in E4 and E23 where continuations shown to be bad for Black are extended without choosing alternative Black moves. The rule claims that the main information that is carried over from past episodes is what kinds of responses have already been considered (together with the current estimate of the worth of the base move).

No assertion is made as to the reasons why certain responses are "normal" and others "unusual," only that the subject has a consistent categorization of moves in such terms. In some sense, the "normal" response is the one which has the highest expectation of being the correct move; and the more "unusual," the lower the expectation. But the subject has no way of computing such an expectation prior to analysis. Furthermore, no assignment of a quantitative expectation is required, only the ordering given by the classification. Thus the categorization is an a priori one, which comes from a blend of personal experience and the publicly available knowledge of good chess play. Nor is the classification used by the subject necessarily correct. Although ignoring attacks completely is invariably somewhat unusual, many good players would consider a counterattack as the "normal" response in many situations.

Figure 8 shows that conformance to R2 is very consistent. Actually, verification depends on the details of move generation within an episode, which will be discussed further there. Roughly, conformance can be checked by noting the brief characterization of the Rework episodes given in parentheses and assuming that the normal-to unusual sequence is: explore, extend, counterattack, counterattack with loss, recapture with loss. Then each Rework episode should be further down this sequence than its predecessor. (Also, the Explore episodes should consist only of normal moves.)

R3 deals with the question of when to change base moves. it says simply, "Stay with a winner, switch off a loser." A single disconfirmation of the soundness of the move is enough to cause the switch. Although changes in base move usually involve generating a new move, R3 does not specify whether one is to obtain a new base move or return to a different old one. Likewise, R3 does not specify at all how the new move shall be selected. In fact, there appears to be little that can be said from this one protocol about how the subject selects base moves.

Of the 21 cases in Figure 8 that are relevant to R3, 16 are confirmatory. Three cases (E3, E6, E14) show that other rules take priority (R5 and R6), and thus shed no light on R3. There two negative instances, E18 and E20. In E18 the subject goes ahead and explores a second variation even though E16 turned out badly. Some light will be shed on this in discussing R4. In E19 the subject has just reviewed the 1.BxN'/5 exchange with discouraging results, but decides to go ahead anyway in E20. Whether one calls this a negative instance or an irrelevant instance depends on whether E19 is viewed as an exploration or only as a summary.

R4 concerns the modification of the search rules in order to get special information. By biasing the choices of moves in favor of White, the subject is able to see if any possibility exists for a successful continuation. If the biased exploration were successful, one would expect additional episodes devoted to correcting the bias; unfortunately, the protocol does not provide good opportunities to test this. A problem posed by R4, and not answered in the rule, is how to bias the opponent's choices without opening the floodgates of foolishness, which would provide no useful information at all. Two hints are provided in the subject's behavior. One is to ignore the opponent's move all together (the "No-move"); this at least leaves open what the opponent might do (E9, E24). The other is to permit the choice from the responses that are normal or almost normal, but which immediate evaluation might not indicate offer the best chance for opponent (E15, E16, E18).

There are only a few cases relevant to R4, but the bias is sufficiently clear to make the rule important. No rule is given to determine when R4 is to be applied to an exploration; consequently, negative instances are not possible. E24 does provide a case where a biased exploration leads to positive results, but no critical followup occurs;

on the other hand, the subject had concluded that the basic continuation (1.BxN/5, P'xB) was favorable to White.

The use of R4 is related to an important feature of the subject's behavior that we have not characterized in the rules. In general the episodes work forward, exploring the consequences of various base moves. In this they agree with the basic philosophy of the chess programs. However, in E15, E16, and E18 a basically different approach is used. In E15 a future situation is envisioned (the mating configuration, B135) an attempt is made to find a sequence of moves that leads to it. As a result of this activity, a difficulty is spotted (B'(K2)) and in E16 the goal is set up of removing this difficulty. Both E16 and E18 are devoted to achieving this goal. The search still works forward, but with a definite end in view. This kind of means-ends analysis is not used in existing chess programs[6]. However, it has been explored in considerable depth in other heuristic programs, particularly in a program called the General Problem Solver (GPS) (Newell, Shaw and Simon, 1960). Humans use such means-ends analysis extensively in other tasks. The condition that appears necessary for its application is that a future condition can be specified in sufficient detail so that relevant differences can be found between the present state and the desired state. To assert in the present position that one wants to obtain a checkmate position does not permit any specific inferences, whereas to say that one wants to get the B at R6 and the Q at N7 lets

[6] Current modifications of the mating combination program mentioned earlier (Simon and Simon, 1962) do include mechanisms of this type.

one go to work. The amount of means-ends analysis in chess would be expected to depend strongly upon whether the positions being considered permitted highly specific future configurations to be envisioned. Three instances of the operation of R4 occurs in these means-ends analysis episodes. Perhaps, having a specific goal in mind is what triggers the need to construct possible continuations that achieve that goal, even if they are not completely realistic.

R5 is a special form of the notion that moves can be considered independently of positions. An exploration may discover new moves to try as well as new facts about the base move. This mechanism, in spite of its plausibility, has not been used in chess programs[7]. Besides the idea of discovering moves in one context and using them in another, R5 also contains the idea of interrupting; i.e., of exploring the new move next. The protocol varies as to whether interruptions can terminate explorations (E18?), side track them (E17), or only obtain priority to be the next episode (E3, E14).

> Of the four relevant cases of R5, all are positive. E3 is completely explicit. E14 is a case of discovering that the N' can recapture the B in E13, and then trying it out at a different place in the 1.BxN'/5 exchange. One might argue that E14 is simply the next variation in the elaboration of the 1.BxN'/5 exchange. The interruptive character of E17 is fully attested to by the return to E16 after E17 is complete. E19, of course, involves the return to an old move, rather than the discovery of a new move. Still, the move,

---

[7] One exception is a program for finding checkmates in two moves (McCarthy, 1959).

> BxN'/5, shows up as the last move of the E18 exploration. There are no negative instances of R5; to have one would require discovering a move (and announcing it in the protocol) and then delaying its exploration for at least one episode.

R6 is the heuristic for looking around when things go well. There is only one positive instance of it, but it is both so clear in the protocol and so important that we record it. E25 may be viewed as a negative instance (and we have so labeled it), since an insistent use of R6 would have required the subject to take one final survey of the whole position before committing himself to the move. The protocol gives no clue on why R6 was evoked after E5, rather than earlier or later.

In summary, if we constructed a program that operated according to rules R1 through R6, using suitable priorities, we would get some of the features of the episodic behavior shown by our subject. These rules are not complete, however. For example, they do not determine how to choose a new base move when switching is called for, when to shift to means-ends analysis, or when to declare a newly discovered move worth an interrupt. Also, they do not determine the internal structure of an episode. This last will be taken up in the next section, but the other questions must remain unanswered.

Move Generation

In this section we wish to construct a move generation scheme to be used at positions within episodes, and to compare the behavior of this scheme with the subject's behavior. We view this scheme as working within the MPD strategy and therefore fitting into the rules of episode generation we have just laid out. To start with we must restrict our attention to those positions in Figure 5 in which move generation takes place. Thus, we exclude the base position and all terminal positions. We also exclude all of E19, the summary, and E25, the recapitulation of the final choice. And we ignore the three "apparent" branchings in E16-E18, E19-E20 and E23-E24, which are due to the subject's not returning all the way to the base position to start the next episode.

Repeated moves. If we now consider Figure 5 with these appropriate restrictions, there are 74 positions in which move generation occurs. We should immediately distinguish those positions in which new moves are generated (53) from those positions in which moves that had already been made are repeated (21). The latter positions appear to pose primarily an issue of whether the position is one from which to start a variation. This need not involve any move generation at all, but only a diagnosis of the position on the basis of the prior analysis and the "instructions" for variation given to the episode. If no variation is to occur, then the subject simply repeats the move made previously.

> Support for this interpretation of these "repeat" positions comes from the fact that in 20 of them only the repeated move is generated. The lone dissenter, in E4, involves a recollection of the Rook exchange in E2 (B50), and it is clear that the move was not considered seriously. In all events, there are no

positions that pose choices between new moves and old moves.

New moves. Considering the positions where new moves are generated, the donimant fact is still that almost always only a single move is generated (43), although occasionally two (9) or three (1). Several features of these positions might provide a starting basis for understanding how a move generation might go. Thus, of the 10 positions with tufts (i.e., with multiple alternatives) 7 are Black and only 3 are White; the corresponding figure for single move positions being 18 Black and 25 White. Thus the subject might be treating himself (White) differently from his opponent (Black). Also noteworthy is the fact that 7 tufts are defensive and only 3 offensive (and the offensive tufts are not all White). The corresponding figure for single move positions is 19 defensive and 24 offensive. It is plausible that multiple moves are generated when on the defensive (and hence constrained),whereas when on the offensive a single aggressive move suffices. However, instead of pursuing either of these possibilities, we will take a different tack that will give us somewhat more specific information.

Move generators for single functions. Existing chess programs generate much larger sets of moves than we require. However, it is not easy to design single integrated processes that will turn out a large collection of plausible moves (e.g., the set of eight moves that the subject considers from the base position). As we indicated earlier in the paper, the solution adopted in chess programs is to develop specific generators devoted to specific functions, and to use higher routines to select these specific generators in order to obtain the total set of moves considered from a position. A cursory examination of the subject's protocol reveals considerable activity of the same sort. For example, B46 says "Black must recapture," thus posing a function to be performed; and B47 follows with "and he can only do it by playing Pawn takes Knight," thus generating the move (presumably the only one) that satisfies the function. If there had been several ways of recapturing, presumably they all would have been generated; and a tuft would have occurred at this position. B23 gives an example where a tuft did occur, the branches all representing recaptures. Although there is good evidence for move generation by function, there is little indication of combining the products of several separate generators. As we have already remarked, almost everywhere only a single move is generated.

This paper is not the place to launch an investigation into the concept of "function" and the full role of function terms in problem solving. We note only that functions operate as intermediaries in the following way. Suppose a piece, Y, is moved so that YxZ becomes possible. We then say "Y attacks Z", which classifies the particular situation. From "attacks" we infer that a problem exists and obtain "defend Z" as a class description of the solution. Under "defend" is available a series of more specific functions that can accomplish this function: "capture attacker," "add defender," "interpose safe man," "pin attacker," "move defendent," etc. At some point of elaboration, we have ways of generating actual moves that accomplish the functions -- e.g., generate all moves that capture the attacker. Thus, we get from problem to solution via a string of functional characterizations, making connections between means and ends of a functional level.

This suggests that we try to specify for our subject the various situations that give rise to recognizable problems and thence to definite generators that provide moves to solve these problems. In any position, if the subject recognizes the situation, he simply generates the set of moves appropriate to a situation of that kind. Only one move may be generated, or more than one; the subject takes whatever the generator produces. However, by implication, all the moves will serve the same function.

We view these specific functional generators as the means whereby rule R2 is carried out. Several move generators may apply to a single situation These are then labeled by the subject "normal," "unusual," etc. On any particular occasion, only one generator will be evoked. R2 asserts they are to be evoked in order.

As discussed in the section on Episode Generation, we view these generators and their labels essentially as public knowledge, although obviously capable of being tinged with the subject's personal experience. For many of the situations English function terms will exist (e.g., "attack"), but there is no reason why there should always be such terms. However, we do expect other chess players of equal (or perhaps somewhat better) caliber to be able to recognize the same problem situations and to know what moves should be proposed to solve them. We will rely on this requirement, that the situations and generators exist in the domain of common chess knowledge, as a check on our creating ad hoc generators to describe our subject. In point of fact, there are no difficulties in interpretation for most of the cases in the present protocol.

Situation-response rules. We give in Figure 9 a list of situations and the responses they invoke. The list includes defensive situations, situations in which the mover has the initiative, and situations where the mover's aim is to acquire information. Responses of this latter type are appropriate for analysis, but not for actual play. Opposite each description is a mnemonic code. This code indicates both the situation (to the left of the vertical bar) and the response (to the right of the bar). Thus the first item is x|r, the "x" standing for the fact that a capture occurred (as in B'xB) and the "r" standing for the response of recapturing. These rules might have been described precisely in the language used for coding the protocol, but this seems superfluous, since the meaning is quite clear. The list of Figure 9 is by no means complete; it contains only those response situations that actually occurred in the protocol. The subject undoubtedly has available many more response schemes than come to light in this particular protocol.

All the rules of Figure 9 are extremely simple and well known in the chess literature. There can be little argument about their general familiarity to someone who plays any amount of chess and who has studied this literature. The information-gathering rules are seldom stated explicitly, of course, Defensive situations

Code	Situation	Response	N
x|r	Man captures last move	Recapture with no apparent loss	11
x|c	Man captured last move	Counter capture of equal value	1
x|x(-)	Man captured last move	Counter capture with apparent loss	2
x|r(-)	Man captured last move	Recapture with apparent loss	1
a|d	Man attacked, by man of not lower value	Add defender	1
a|xet	Man attacked, by man of not lower value	Exchange target	8
a|ea	Man attacked, by man of lower value	Exchange attacker	1
a|m	Man attacked, by man of lower value	Move target away	5
a|c	Man attacked	Counter attack of equal value	1
2a|2d	Double attack	Add double defender	1
Initiative situations			
iP|a	Opponent P isolated	Attack P	2
p|a	Opponent man pinned	Attacked pinned man	1
e|ed	Opponent exchange, just defended	Exchange defender	1
e|mde	Opponent exchange just defended	Move defender away by forcing its use in another exchange	2
e|x(+)	Opponent exchange, under defended	Capture target with gain	2
Information gathering situations			
e|g	In midst of exchange	Go on exchanging	1
c|g	Counter attacked	Go on with primary attack,	1
g|g	In midst of plan, no threat exists	Go on with plan	11
|n	Assertion made about situtation	Take action, assuming negation, to test assertion	3

Legend

a	attack	g	go on	p	pin	(-) with loss
c	counterattack	i	isolated	r	recapture	(+) with gain
d	defend	m	move	x	capture	
e	exchange	n	negation			

Figure 9. Situation-response move generators

since they are part of "common sense" of everyday living. However, some caution is indicated in the case of g|g, which implies that a plan is operating, from which decisions about continuing the plan are derived. Before g|g can be inferred we must infer the plan.

Figure 10 gives for each of the 64 new moves the mnemonic code of the rule from Figure 9 that appears to govern its generation. The figure permits an assessment of the extent to which our set of simple situation-response rules can account for the behavior of the subject. Overall, one might say that in the majority of positions (39) the account is reasonably good; and for an appreciable number (14) there is some reason to be dissatisfied. But this obscures the great variety of ways in which the rules can fail to account for move generation. Also, some of the "failures" can be explained, and some of the "successes" are not as solid as they appear on gross tabulation. Consequently, a certain amount of detailed treatment of the data is necessary.

Incomplete generation. If the situation at a position evokes a rule, we expect all moves generated by that rule to be considered. Where this is not the case we have marked the position with an "i" (for incomplete). However, for several of these, there is clear evidence in the protocols that the subject believed that he had generated them all. In these cases it is more reasonable to argue that the application of the rule was faulty, rather than that the overall scheme does not account for the subject's behavior. There still remain seven cases of incompleteness. These appear to be genuine failures. They have about them the aura either of additional considerations, not present in the

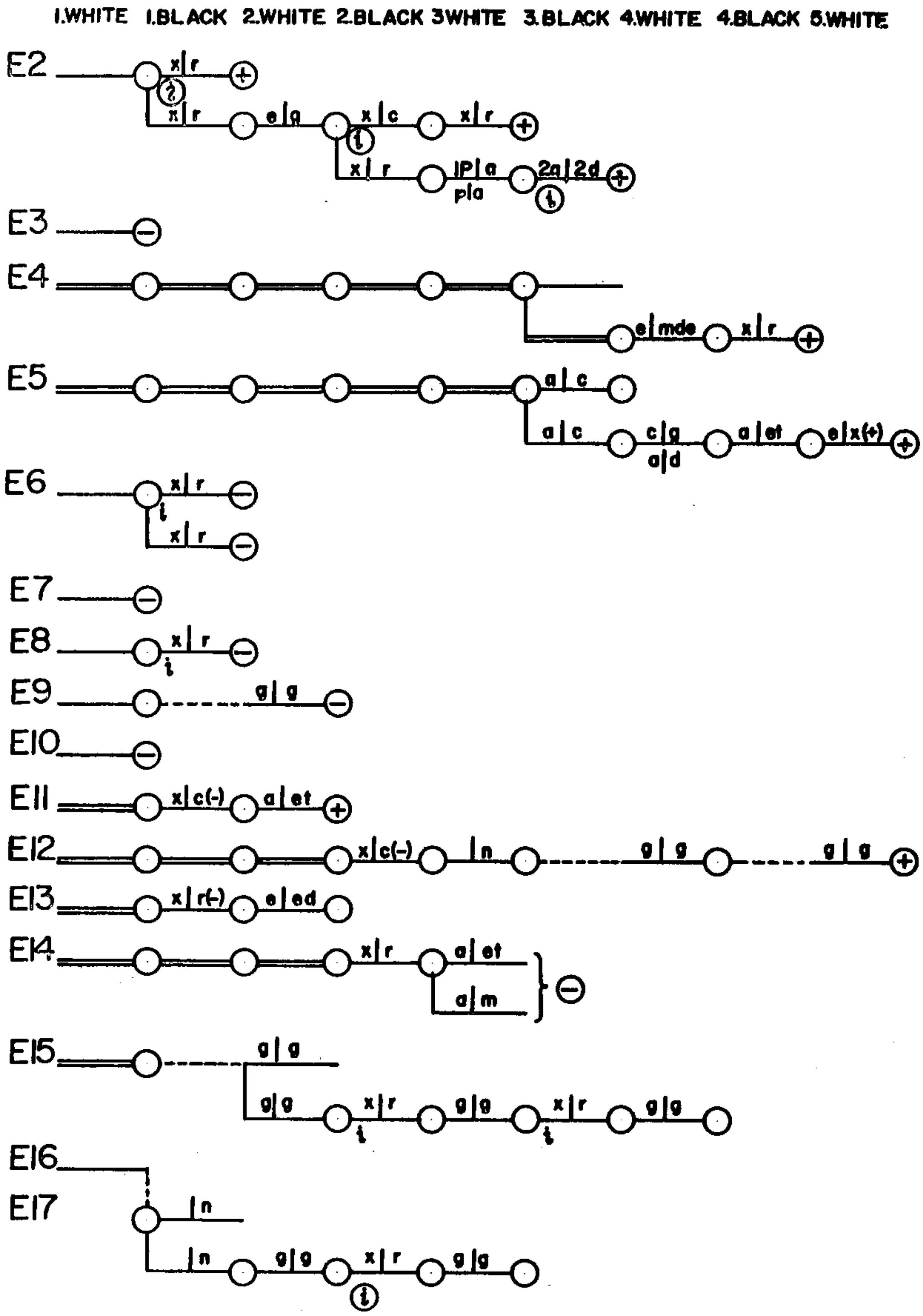

Figure 10: Verification of move generators

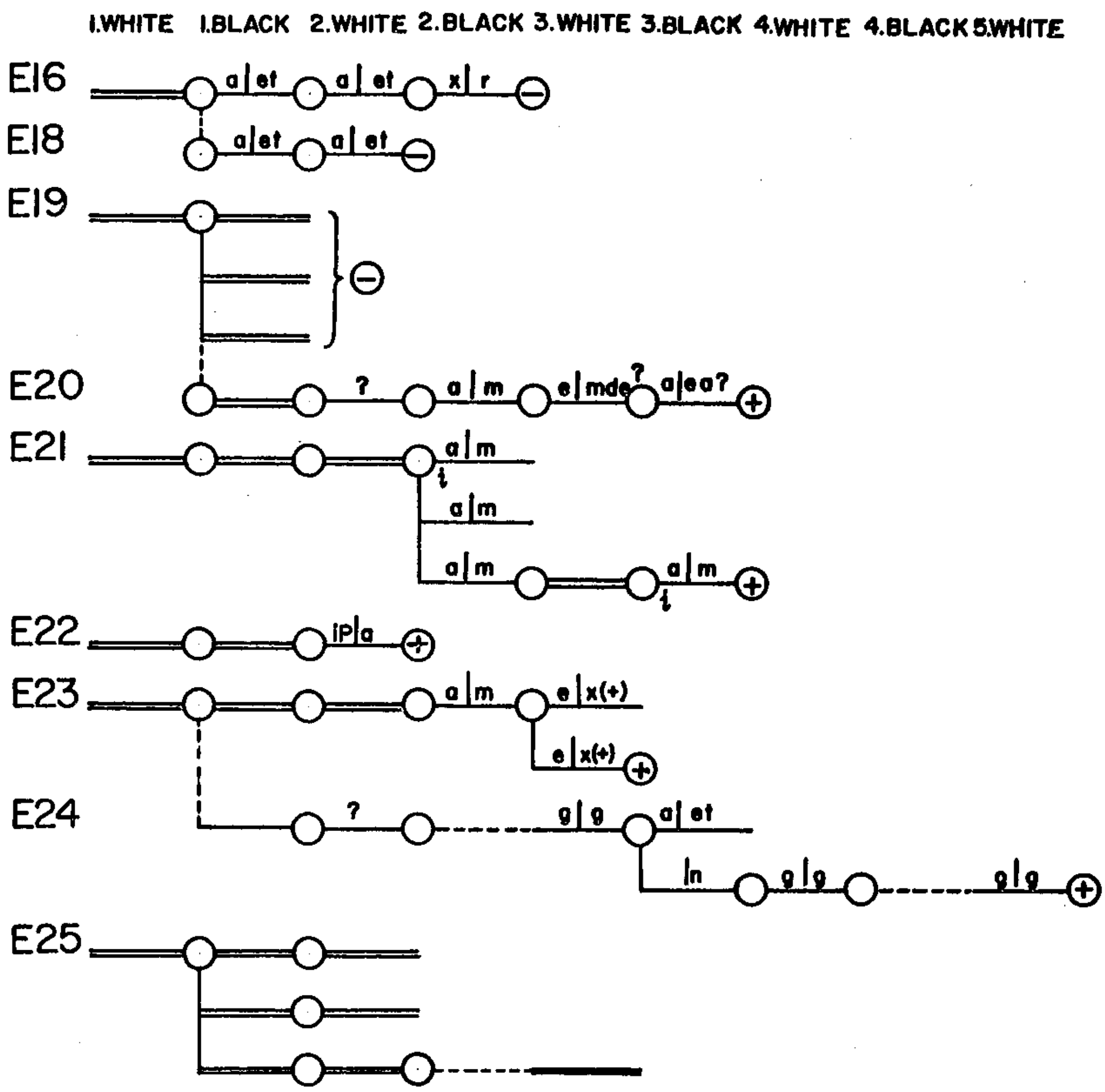

Figure 10: Verification of move generators
(continued)

simple rules of Figure 9, or of the requirement only to demonstrate sufficiency, so that a single successful candidate will do.

There are four instances where the subject appears to believe his generation is complete, although objectively it is not. In E2, he ignores 1...N'xB, declaring the N' to be pinned, which it is not. This error is repeated at the next move where he ignores 2...N'xN. Also in E2 the subject asserts that 3...Q'-Q1 is the only move that simultaneously defends both attacked men, but two other Q' moves are also possible. In E17 the subject seems temporarily unaware of 2...R'xN since he ignores 3...R'xR a move later.

There are three cases (E6, E8, E21) where incomplete generation occurs on the final move (i.e., at the postion before the last one); in all these there is no need to do more than obtain a "typical" move to provide the evaluation. There are two cases (E17, E21) where not all the Q' moves are generated -- e.g., Q'-R3. These could have been eliminated by additional reasoning, or they could be a failure of the generator. Finally, there are two cases in E15 which are intimately tied up with rule R4 and biasing search in favorable directions.

More than one generator. To account for the small number of alternatives generated at each position, our scheme posits that only a single rule is evoked. Positions in which only a single move are generated clearly conform to this. However, the tufts can provide counter-examples, and two cases out of ten show some evidence of multiple generators being evoked. Note that our one-position-one-generator rule is somewhat ambiguous, since functions form a hierarchy. For example, the function of defense can be realized by adding a defender or moving away, and these might be functional equivalents as far as the subject is concerned.

One case of mixed generators occurs in E2 where 2...P'xN is clearly a recapture (x|r), whereas the other is clearly a counterattack (x|c). This may be a case of interruption (rule R5). The subject shows a tendency to attend to the possibilities uncovered by a move (e.g., E17), and 2...R'xR is possible only because of 2.NxB'. A second case occurs at the end of E14. Here, two (perhaps equivalent) defenses for the B are considered. Note that one of them is a generalized move. A parallel situation, but one that does not quite generate an actual move occurs in E5 at 4...B'xB, where the subject remarks that Black must either recapture or defend his Bishop (B72). In one other tuft (E24) there are a pair of moves which are not functionally equivalent (2...P'xP and 4... not (P'xP)). But, as discussed below, these are clearly generated by a single rule (|n).

Multiple interpretation. It is possible for more than one rule to account for a single generated move. However, it happens in only two cases here. The principle of making moves that serve more than one function is important and well known in chess. From an information processing viewpoint we would expect such a move to be generated from one rule and then recognized as meeting the requirements of another. However, the rule 2a|2d incorporates directly finding a double function move. In any event, we should not treat cases of ambiguous interpretation as showing serious deficiences in how our scheme accounts for the subject's behavior.

> One case of multiple interpretation occurs in E2 at 3.Q-B3. This can be generated by iP|a, and the repetition of this function in E22 (B202 and B205) reinforces this interpretation. However, the move can also be generated by the rule of a pinned man (p|a), and it is clear that the subject considers the double threat (B34, B35). The second case of multiple interpretation occurs in E5 after the double threat, 3.Q-B3, and Black's counterattack, 3...Q'xQP. The move, 4.BxN', can be seen both as a way of adding a defender to the N'(K5) (a|d), which the subject recognizes is in danger (B69), or as continuing the original attack to see what follows (c|g).

No generator. For two moves it does not seem possible to assign any of the situation-response rules, and for two others the assignment is quite uncertain. We have noted these moves with a question mark (?).

The entire E20 episode is somewhat obscure. The subject needs an alternative White move, but his grounds for choosing 2.N-R4 are not evident. Likewise the follow-up, 3.N-B5, is perhaps an attempt to dislodge the defending B'(K2), since he apparently believes he is attacking the Q'(B193). The other unassigned move is 2.P-B4, which is clearly a move connected with initiating a K-side attack, but not one where it was possible to identify a familiar configuration that could determine the move.

Generation of plans. There are several situations where the subject has developed a plan of action that implies a sequence of moves. The moves generated at eleven positions can be accounted for on this basis (g|g). When implementing a plan the subject seems to operate in a very exploratory way. Thus, all the positions where the opponent's move is skipped (the "No-move") occur during these times, and the subject almost never generates alternative moves for implementing a plan. Some of the plans are so familiar that they could have been made into situation response rules -- e.g., doubling the Rooks in E9. Others, such as the means-ends analysis of E15, are clearly constructed for this particular situation. The technique of attempting to demonstrate a fact by assuming its negation and following out the consequences, is used sufficiently often that we have made it a rule (|n). This technique generates a plan since the next several moves are dictated by the attempt to elicit the pertinent consequences.

There are six instances of planning. Given a plan, the derivation of moves is easy. The plan of E9, to double the rooks on the QB-file, is not too clear in the protocol. In E12 the attempt to show that the B is not pinned by showing that the N is not pinned is clearly stated. The means-ends sequence of E15 has already been discussed. In E17 the subject is concerned with determining whether the Q' has been tied down by the double attack on the B'(B3), and he does this by exploring the consequences of letting the Q' capture the NP. The final two plans occur in E24. The exploration of the Pawn push of the King side (2.P-B4) is clear enough, although its origin is obscure as we have mentioned. The other plan is the attempt to determine the consequences of Black not doing 3...P'xP, and is clearly stated (B228).

Ordering of generators. In many positions more than one situation-response rule is applicable. This choice of generator is one means by which rule R2, which describes the sequencing of episodes, is carried out. In Figure 9 the situation-response rule are given in order. Thus the normal response to a capture is to recapture (x|r), after which (for the subject) comes a counter capture (x|c), and then actions that entail loss. In many of the situations the protocol provides no opportunity for a series of responses -- e.g., the initiative situations. From the protocol it appears that at each episode that reworks a previous episode the prior path is followed until a new situation-response rule is applied to generate a variation. From this point on the subject carries out the search as an initial exploration, since all the subsequent positions are new.

Summary of move generation analysis. The scheme of Figure 9 is appealing in its simplicity. It says that the subject, through prior experience with chess and chess literature, has an available collection of fixed responses to specific situations. Behavior directly follows recognition. The evocation of only a single generator and the specificity of the chess position itself yields the fact that the number of alternatives generated at a position is one or at most a few moves. The fact that most tufts are both Black and defensive is to be explained as a derivative characteristic of the generators that are appropriate to this particular base position. Our attempt to verify this scheme in Figure 10 and the subsequent discussion shows both that the scheme can account for an appreciable number of move generations, and that detailed consideration of each instance is required to make sense of the evidence.

Each item in Figure 9 is based on a small exercise in logic -- e.g., the idea behind defending an attacked man is that if he is defended, the opponent can no longer make a gain. In each instance in Figure 10 one could argue either that the subject has a situation-response scheme, as we have outlined it, or that he brings to bear a more general problem solving mechanism to discover the same moves that the scheme gives. In general, the scheme seems more probable when the generated moves appear in the protocol immediately upon occurrence of the situation without evidence of substantial problem solving; but each instance must be argued on its own merits.

Evaluation

There remains the question of when an episode terminates and what evaluation, if any, is assigned. The fundamental logic of the game -- that one maximizes for self and minimizes for the opponent -- is clearly used throughout. The subject operates as if permanent gain is unlikely from any position, so that any pernament gain can be used to terminate an episode. Thus, any gain that a player has when it is still his opportunity to move can be considered permanent. Likewise, any loss that exists for a player at his move, for which there does not exist immediate compensation, can be considered permanent. This latter occurs especially when a capture has just happened for which there is no immediate recapturing possibility. The lapses from these terminating rules occurs most often with deliberate counter attacks with loss and recaptures with loss. As we have seen, these are unusual responses and only occur in latter reworkings of an episode.

The dominating characteristic of the subject's evaluations is that they involve only a single feature of the position -- e.g., a Pawn isolated. At the end of each move (i.e., a move for White followed by one for Black) either the status quo is preserved, in which case search continues, or an advantage one way or the other exists, in which case the search terminates. Since almost always there is a single cause for the change in evaluation, this stands as the single evaluative feature. There is no assignment of an evaluation on the basis of balancing a pro from one feature with a con from another. Thus the form of interaction implicitly provided for in the polynomial payoff functions of chess programs, and capitalized on when learning is attempted on the polynomial weights, does not appear. As we remarked earlier, this behavior agrees well with that of de Groot's subjects.

Figure 11 shows the various evaluations that are used by the subject. Opposite each are the statements in the protocol where these can be found. They are not listed in any order. There are no occasions when more than one term appears in an evaluation. Balancing of material does occur -- e.g., a piece for a pawn. In a few instances there is a listing of specific conditions involved in preserving the status quo in addition to stating the advantage or disadvantage of the move.

Most of the terms in Figure 11 are common chess terms with standard operational meanings -- "isolate Pawn," "lose tempo," "obtain open file," etc. The subject uses only a few terms that are vague -- "complications," "mess up," "pressure." In this respect he differs from some of de Groot's subjects, who abounded in such general evaluations as "N-K4 -- take it away!" With such phrases one can still search for an underlying global, impressionistic, Gestalt-like evaluation. With our subject this is somewhat more difficult to do.

Closely allied to evaluation is the use of levels of aspiration -- controlling search by setting a threshold such that only changes in the evalution of the position that exceed threshold are used to terminate the position. Search starts with the threshold set at some reasonable level; if positions are obtained continually that exceed threshold, the aspiration level is raised;

Win material	B54, B76, B87, B97, B103, B218
Obtain no threat	B39, B85, B115, B123, B133, B158, B165
Remove threat	B36
Obtain open file	B28
Isolate Pawn	B25, B202
Double men on file	B81, B83
Lose tempo	B89, B93, B170, B197
Put Queen out of play	B66, B75
Make retreating move	B190
No way to get Q on N7	B146
Make man hard to defend	B206
Remove control from square	B222
Lead to complications	B175
Put on pressure	B181
Mess up K'side	B230

Figure 11. Types of Evaluations

if few positions are found that exceed threshold, the level of aspiration is lowered. With the evaluative procedure that our subject is using, it is difficult to see such a mechanism in operation. One striking example of it occurs in the scene where the subject looks for new possibilities (E6 - E10). The moves generated during this period are much poorer than the moves appearing elsewhere (e.g., 1.NxNP'). It appears as if the bars had been let down in order to gather in all the possibilities.

Conclusion

With only a single protocol at hand we would not claim any universality of the picture we have drawn of human chess play, either for this particular subject or for human chess players generally. Although we have treated several matters in some detail, there are others which have been ignored entirely or only hinted at. A listing of the full set of parts required to compose a program according to the MPD strategy, analogous to our listing of the parts of current chess programs, would reveal how many things are still unspecified. For example, we have given no move generator for base moves, no criterion for interruption, no rule for when to apply a new situation-response rule when reworking an episode; and no rule for when to terminate the entire analysis. There may be levels of organization we have not touched; say at the level of the seven scenes we used in describing the entire protocol. Clearly, a single organizing idea underlies all of the "Search widely" scene (E6 - E10).

The subject's analysis of the position was not perfect. At least two important features of the position apparently were never considered. The

B'(K2) is "hanging," and thus is an appropriate weakness to exploit. Also, N-Q7 forks the Q', the R'(KB), and the B'(K2) if it recaptures on KB3. This threat provides the real refutation of 1...B'xB after 1.BxN'/5. Besides these errors, the subject made many minor errors in generation, perception, etc., some of which have been alluded to in discussing move generation. We have not made any exhaustive analysis of these imperfections, nor provided any model for why they should have occurred.

We have tried to show that a human's analysis of a chess position can be understood in the same terms we use for chess programs and other problem solving programs -- i.e., as an information processing system. Although the search strategy in the protocol is different from those used to date in chess programs, our subject does have a consistent strategy and one that could be incorporated into a program. Similarly, the subject's evaluations of terminal positions of search is not a polynomial such as is used in most chess programs. It is much simpler and less subtle (although formally it is a special case of a weighted polynomial). The kinds of generators the subject uses are mostly of the same genre as those used by programs -- i.e., based on simple functions to be performed. However, they are in some cases more subtle than those used in present programs. And at higher levels of integration, the subject engages in means-ends analysis, working backwards from a desired goal to set up a subgoal. These higher levels of organization are missing from chess programs, although the general means-ends analysis forms the basis of other problem solving programs.

In short then, we see in this one subject's behavior in this one analysis, an information processing system that is different in organization, but similar in componentry, to present problem solving programs.

References

Bernstein, A., Roberts, M. de V., Arbuckle, T. and Belsky, M. A., (1958); A chess playing program for the IBM 704. Proceedings of the Western Joint Computer Conference, 157-159.

Bruner, J. S., Goodnow, J. J. and Austin, G. A., (1956): A Study of Thinking, New York. Wiley.

de Groot, A. (1964): Thought and Choice in Chess. The Hague: Mouton. (Revised translation).

Feldman, J., (1963); Simulation of behavior in the binary choice experiment. E. Feigenbaum and J. Feldman, Editors. Computers and Thought. New York. McGraw-Hill. (p.329).

Green, B., (1963): Digital Computers in Research. New York. McGraw-Hill.

Kister, J., Stein, P., Ulam, S., Walden, W. and Wells, M., (1957); Experiments in chess. Journal of the Association for Computing Machinery. 4, 2, 174-177.

Kochen, M., (1961); Experimental study of 'hypothosis formation' by computer. C. Cherry, Editor. Proceedings of the Fourth London Symposium on Information Theory. New York. Academic Press. (p. 377).

Kotok, A., (1962): A Chess Playing Program for the IBM 7090. Bachelor's thesis, Massachusetts Institute of Technology. (This program was the joint work of several students at MIT, including besides A. Kotok, C. Niessen and M. Lieberman.)

McCarthy, J., (1959); private communication.

Newell, A., Shaw, J. C., and Simon, H. A. (1958); Chess playing programs and the problem of complexity. IBM Journal of Research and Development, 2, 4, 320-335.

Newell, A., Shaw, J. C., and Simon, H. A., (1960); Report on a general problem solving program for a computer. Information Processing: Proceedings of the International Conference on Information Processing. Paris. UNESCO. 256-264.

Newell, A., and Simon H. A., (1963); Computers in psychology. R. D. Luce, R. Bush and E. Galanter, Editors. Handbook of Mathematical Psychology, vol. 1. New York. Wiley (p. 361).

Samuel, A., (1959); Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3, 3, 210-229.

Shannon, C., (1950); Programming a digital computer for playing chess. Philosophical Magazine, 41, 356-375.

Simon, H. A. and Simon, P., (1962); Trial and error search in solving difficult problems: evidence from the game of chess. Behavioral Science, 7, 4, 425-429.

Wiener, N., (1948): Cybernetics. New York. Wiley (First edition).

Appendix: The Protocol

Each line of the code attempts to express what information the subject must have been attending to in order to have made the corresponding utterance. Thus, each line could be prefixed by the phrase "The subject considers...."

The general conventions of the code are familiar from English and standard mathematical notation. The expressions are compounded, in the usual way, of relations (e.g., N threaten Q'), properties (e.g., P' isolated), functions (e.g., Number (P)) and action phrases (e.g., Find move for White). With one or two exceptions given below the various terms are used in a sense close to their meaning in English or in chess, and no special definitions need be given for them.

Standard chess notation for moves is used. This is extended slightly by always priming the Black men and by permitting generalized moves to be used. Thus, "1.BxN'/5" means the Bishop at Rook 2 captures the Knight on the fifth rank on White's first move; "3...Q'-move[back]" means a backward move of the Black Q on Black's third move; "1.Move" means simply White's first move. The only other new terms are "Pc" for "piece" (i.e., a man other than a Pawn) and "[threaten Black]" for "that which threatens Black."

To verify the relations between the various moves mentioned by the subject it is desirable to show in the code each bit of explicit evidence from the utterance that one move follows another or is a response to another. However, for readability, if a move follows the one mentioned in the line preceding it, we write down only the move. Only if there is some intervening discussion, do we explicity mention what move is being followed. We always mention when the subject indicates a response, since this is a stronger connection between moves than merely that one follows the other.

A few notes are appended at the end of the protocol, as indicated by asterisks (e.g.,B2*).

As an aid to gaining an appreciation of the code we list the various terms that occur in the coding.

Chess terms:

Sides:	White, Black
Men(M):	K, Q, R, B, N, P, Pc
Board:	K-side, Q-side, file, rank, behind, on, at
Moves:	follow, response to, next move after, after, retreat, back, path
Others:	attack, bear on, checkmate,defend, destroy, effective, escape, exchange, fork, isolated, lose, open file, pin to, remove, safe, structure, tempo, threaten, win

Non-chess terms:

Quantifiers:	all, exist, many, most, only
Connectives:	and, not, or
Actions:	add, choose, confirm, consider, find, get, repeat
Means-ends relations:	Against, correction to, for, necessary, possible, sufficient
Others:	new, number, result, set, => (produces or causes)

E1	B1	Side(1.Move) = White	Okay, White to move...
	B2*	Set(Pc) = set(Pc')	In material the positions are even.
	B3	Number(P) = number(P') = 6	One, two, three, four, five, six-six Pawns each.
	B4	Find [threaten White]	Black has what threats?
	B5	Q' threaten NP	His Queen is threatening my Knight's Pawn
	B6	Q' bear on QP	and also he has one piece on my Queen's Pawn -
	B7	QR' behind B'(B3)	has a Rook in front of the Bishop,
	B8	B7 => open file possible	which will give him an open file.
	B9	Find [threaten Black]	Let's see, all right, what threats do we have?
	B10	Only B(N5) attack N'(B3)	We have his Knight under single attack
	B11	B'(K2) defend N'(B3)	protected by the Bishop.
	B12*	N(B3) attack N'(Q4)	We have his other Knight under attack
	B13	(KP' and N'(B3) and B'(B3)) defend N'(Q4)	protected by three pieces.
	B14	Q bear on KNP'	The Queen is bearing down on the Knight's Pawn
	B15	R(QB1) defend N(B3)	and the Rook is over here protecting the Knight
	B16*	B(R2) bear on N'(Q4)	and the Bishop at Rook 2 is bearing down on the Knight.
	B17	White attack K'side possible	All right, looks like we have something going on the King's side.
	B18	All Pc' on Q'-side	All Black's pieces are over on the Queen's side -

	B19	Most Pc'(B18) not effective for B17	most of them out of play -
	B20	B17 possible	good chances for an attack perhaps.
E2	B21	Find 1.Move	See, what moves are there?
	B22	1.BxN'/5; possible	The Bishop at Rook 2 can take the Knight,
	B23	(1...B'xB or 1...P'xB); response to B22	which would be no doubt answered by either Bishop takes Bishop or Pawn takes Bishop.
	B24	1...B'xB desirable for Black	Probably Bishop takes Bishop
	B25	1...P'xB => P' isolated; => B24	to avoid isolating the Pawn.
	B26	2.NxB'; follow B24	If we then play Knight takes Bishop,
	B27	(2...P'xN or 2...R'(QB)xR); response to B26	he will then play Pawn takes Knight or Rook takes Rook,
	B28	2...R'xR => exchange R => open file for White	but this would give White an open file if he exchanged
	B29	Result(B28) not desirable for Black	and this is doubtful.
	B30	2...P'xN => P' isolated	This would isolate Black's Queen's Pawn -
	B31	(Only N'(B3)) defend P'	it would be protected only by the Knight
	B32	B(N5) pin N'(B3) to B'(K2)	which is pinned,
	B33	3.Q-KB3; follow B30	therefore we could move the Queen to Bishop 3,
	B34	B33 => add (Q threaten N')	not only putting another threat on the Knight,
	B35	B33 => Q threaten P'(Q4)	but also threatening an isolated Pawn.
	B36	(Only 3...Q'-Q1) defend N' and P'	Both of them could not be protected simultaneously unless Queen to Queen 1.

E3	B37	1.Q-KB3	All right, well, what about Queen to Bishop 3 immediately.
	B38	B37 not desirable	Queen to Bishop 3 immediately is not good -
	B39	B37 not => [threaten N'(B3)]; => B38	it gives no threat on the Knight at Bishop 3
	B40	N'(Q4) defend N'(B3) => B39	because it is protected by the Knight at Queen 4.
E4	B41	Repeat consider path(1.BxN'/5)	So let's follow this through again.
	B42	1.BxN'/5	Bishop takes Knight
	B43	1...B'xB; response to B42	which will be answered by Bishop takes Bishop.
	B44	2.NxB'	We will play Knight takes Bishop
	B45	B44 => N threaten Q'	threatening the Queen -
	B46	2...(Black)xN necessary against B45	Black must recapture
	B47	(only 2...P'xN) for B46	and he can only do it by playing Pawn takes Knight.
	B48	3.Q-KB3	Then if we play Queen to Bishop 3,
	B49	3...Q'-Q1 necessary against B48	Black is forced -
	B50*	Exchange R possible; => (B49 not necessary) possible	oh, I was forgetting about the exchange of Rooks -
	B51	B49	Black is forced to play Queen to Queen 1.
	B52	3.RxR'; follow B49	If then we exchange Rooks,
	B53	B52 => 3...Q'xR necessary	Black must take the Rook with the Queen
	B54*	B53 => White win P'	and we would be able to win a Pawn safely.
E5	B55	Repeat consider path(1.BxN'/5)	However, I'll just go through again.

B56	1.BxN'/5	Bishop takes Knight,
B57	1...B'xB	Bishop takes Bishop,
B58	2.NxB'	Knight takes Bishop,
B59	2...P'xN	Pawn takes Knight -
B60	3.Q-KB3	Queen to Bishop 3,
B61	(Black response to B60) exist	White has the answer - Black has the answer there.
B62	(3...Q'xNP or 3...Q'xQP) for B61	Queen takes Knight's Pawn if he wishes or Queen takes Queen's Pawn.
B63	3...Q'xNP not difficulty for White	Queen takes Knight's Pawn is no trouble
B64	Q defend RP against Q'xRP; => B63	because our Rook's Pawn is protected by the Queen
B65	Only (Q' bear on RP) after move(B63); => B63	and he has nothing else down there,
B66	Q' not effective after B63; => B63	he's just putting his Queen farther out of play.
B67	3...Q'xQP (B62)	Queen takes Queen - takes Queen's Pawn...
B68	B67 => difficulty for White	is a little worse
B69	B67 => Q' threaten N(K5); => B68	Because then he's threatening our Knight.
B70	4.BxN' response to B67	So we ... so if we answer that by Bishop takes Knight,
B71	Find 4...Move	he follows with what - he follows with,
B72	B71 => only (4...B'xB or 4...Move[defend B'])	well, he must either take the Bishop or protect his Bishop at King 2.
B73	5.QxB'; response to 4...B'xB (B62)	If he takes it we answer it with Queen takes Bishop.

	B74	B73 => Qdefend N(K5)	therefore - thereby protecting our Knight at King 5
	B75	B73 => Q' not effective	and leaving Black's Queen out in the cold
	B76	B73 => White win Pc' and lose P	and we have won a piece for a Pawn.
E6	B77	Find 1.Move[new]	Let's see if there's anything else here.
	B78	1.NxB'; for B77	Our Knight at King 5 can take the Bishop immediately,
	B79	B78 not desirable	but this - this hardly seems good -
	B80	1...Q'xN; follow B78	Queen takes Knight,
	B81	B80 => Q' and R' on QB-file	then gives him two pieces on the file
	B82	1...R'xN; follow B78	or Rook takes Knight
	B83	B82 => 2 R' on QB-file possible	allows him to double up Rooks on the Queen Bishop file.
E7	B84	1.NxBP'; for B77	The Knight at King 5 can take the Pawn at Bishop 2,
	B85	B84 not => [threaten Black]	but this does not lead to any threat-
E8	B86	1.NxNP'; for B77	can take the Pawn at Kinght 3 -
	B87*	1...BP'xN response to B8; sufficient	this is easily answered by Bishop's Pawn takes Pawn.
E9	B88*	Get R(QB) and R(KB) effective on QB-file; for B77	Both of our Rooks,
	B89	2 move necessary for B88	both of the Rooks cannot get into play more than two moves
	B90*	B89 => B88 not possible	so they're out of the picture temporarily.
E10	B91	Get P(K-side) attack K'side; for B77	A King side push of Pawns
	B92	B91 for destroy K'side	to break up Black's King side

	B93	Many move necessary for B91; => not desirable	would take too long,
	B94	White defend QNP necessary; => B93	because we are after all under the necessity of protecting the Queen's Knight Pawn
	B95	Black attack QP possible; => B93	and also watching out for an attack on the Queen's Pawn.
	B96	B77 not => move[desirable]	So, therefore
	B97	B96 => (1.BxN'/5 if => win Pc' and lose P)	the immediate exchange seems indicated if we can win a piece for a Pawn.
E11	B98	1.BxN'/5	All right - starts out with the Bishop at Knight 2 taking the Knight.
	B99	1...(Black)xB necessary against B98	Black must recapture
	B100	Not B99 => Black lose Pc'	or else he's lost a piece.
	B101	1...Q'xNP; follow B98	If he plays Queen takes Knight's Pawn,
	B102	2.BxB'	then we can play Bishop takes Bishop if we wish
	B103	B102 => White win Pc' and lose (1 or 2)P	and we will come out a clear piece ahead for a Pawn or two.
E12	B104*	1.BxN'/5	Let's see, now, Bishop takes Knight, Bishop at Knight 2 takes a Knight
	B105	1...Q'xNP	followed by Queen takes Knight's Pawn.
	B106	2.BxB'	Then we play Bishop takes Bishop we'd say,
	B107	2...Q'xRP	then Black can play Queen takes Rook's Pawn,
	B108	B107 => B(B6)-move not possible	and thus we cannot move our Bishop at Bishop 6
	B109	B110 and B111 => B108	because if we did that

	B110	Move(B108) => 2Pc' bear on N(B3)	we would put two pieces on our Knight at Bishop 3
	B111	Move(B108) => (Q' pin N to Q) and (R'(B1) pin N to R)	which would be pinned in an attack by the Queen and the Knight, Queen and the Knight - Queen and the Knight - Queen and the Rook at Bishop 1 simultaneously.
	B112	2 M defend N(B3) against B110	So, it's pinned, however, it's protected twice
	B113	Remove pin(B111) possible	and we can break the pin
	B114	Q-move[back] and N-move follow Q-move; => B113	by moving the Queen back and then moving the Knight
	B115	(B112 and B113) => B107 not threaten White	so that is not a serious threat.
E13	B116	1.BxN'/5	So we play Bishop takes Knight -
	B117	1...(Black)xB necessary against B116	Black must recapture,
	B118	Not B117 => White win (1 or 2) Pc' and lose 2 P	if he doesn't he'll lose a piece or two for a couple of pawns.
	B119	Not 1...N'xB for B117	He will not recapture - he will not recapture with the Knight -
	B120	1...N'xB possible; correction to B119	yes, he can recapture with the other Knight.
	B121	1...N'xB; follow B116	If he recaptures with the other Knight,
	B122	2.NxN'	we would of - we would play Knight takes Knight
	B123	B122 => B(N5) safe	therefore our Bishop at Knight 5 is immune
	B124	N attack Q' => B123	because his Queen is attacked.
E14	B125	1.BxN'/5	Okay, Bishop takes Knight

	B126	1...B'xB	followed by Bishop takes Bishop.
	B127	2.NxB'	We then play Knight takes Bishop.
	B128	2...N'xN possible	Again, Black can recapture with the Knight -
	B129	B128 new	this was overlooked.
	B130	2...N'xN; follow B127	All right, Black recaptures with the Knight -
	B131	3.BxB' or 3.B-move[retreat]	what do we have? The Bishop must either capture or retreat -
	B132	B131 not desirable	there, we do not have very much.
	B133	Path(1.BxN'/5) not desirable	So this exchange variation doesn't win us anything.
B15	B134	Structure(K'side) not desirable for Black	Now, Black's King side is in sad shape -
	B135	Get (B(N5) at R6 and Q at N7) => checkmate(Black)	There is a mate if we can get the Bishop down to Rook 6 and sneak the Queen in at Knight 7.
	B136	Find path for B135	So, how do we do this?
	B137	1.Q-KB3; for B136	An immediate Queen to Bishop 3.
	B138*	2.NxN'/5 next White move after B137	Ah, lets see, we will play Knight takes knight
	B139*	2.BxN'/5; correction to B138	play Bishop takes Knight. Bishop takes Knight -
	B140	2...N'xB	Knight takes Bishop.
	B141	3.NxN'	Then where do we stand - then we play Knight takes Knight
	B142	3...P'xN	and Black will play Pawn takes Knight.
	B143	Find 4.move	Then - then what do we play?
	B144	4.B-R6 for B143	We play Bishop to Rook 6.
	B145	B144 => K' escape not possible	If we play Bishop to Rook 6 we have the King trapped down there,

	B146	(Get Q at N7) not possible	but there isn't any way to get the Queen -, the Queen down into Knight 7
	B147*	B'(K2) => B146	because - because of the Bishop at Queen 2.
	B148*	Remove B'(K2); necessary for B135	Therefore it's necessary to get rid of the Bishop to Queen 2 before we can do anything for a mate.
E16	B149	Get B148	All right, the Bishop at Queen 2 -
	B150	1.N-K4; for B148	let's consider the move Knight to King 4.
E17	B151	B150 => 2 Pc bear on B'(B3)	Knight to King 4 puts a couple of pieces on the Bishop at Bishop 3
	B152	B150 not pin Q' to defend B'(B3)	and well, it doesn't really pin the Queen
	B153	1...Q'-Move[back] and 1...Q'xNP; for B152	Because the Queen has got ... the Queen can go back and the Queen has Knight takes Pawn,
	B154	1...Q'xNP(B153) => Black win P and lose P'	which would get back the Pawn we'd win
	B155	2.NxB'	if we played Knight takes Bishop,
	B156	2...P'xN	Pawn takes Knight,
	B157	3.RxP'; => P'(B154)	Rook takes Pawn -
	B158	B151 not desirable.	no, we don't have anything there.
E16	B159	1.N-K4; repeat B150	All right, but Knight - Knight to King 4
	B160	B159 => 2 Pc bear on N'(B3)	puts two pieces on the Knight - two pieces on the Knight at Bishop 3.
	B161	1...N'xN; follow B159	If he plays Knight takes Knight,
	B162	2.BxB'	we play Bishop takes Bishop.

	B163	2...N'xB response to B162; sufficient	This is easily answered by - Bishop takes Bishop - this is easily answered by Knight takes Bishop
	B164	B163 => 3.QxN' not possible	and then we cannot take the Knight which has - which is at Black's King 5.
	B165	B164 => B159 not desirable	So that move seems to be fruitless.
E18	B166	B(R2)-move necessary	We have to get the Bishop out of Rook 2
	B167	Consider not B(R2)-move	because if we do not get it out of Rook 2 - yeah, if we don't get the Bishop from Rook 2,
	B168*	1...Q'xNP	Queen takes Knight -
	B169	B168 => B166	forces us to move it
	B170	B169 => White lose tempo	thereby losing a move.
	B171	(Only BxN'/5) desirable for B166	The only place the Bishop can go with any sense is to take the Knight,
	B172	B171 => 1.BxN'/5	so let's take the Knight right away.
E19	B173	1.BxN'/5	Takes the Knight -
	B174	1...P'xB or 1...B'xB or 1...N'xB	then he can play Pawn takes the Knight, Bishop takes Knight or Knight takes - play Pawn takes Bishop, Bishop takes Bishop or Knight takes Bishop.
	B175*	All move(B174) => difficulty	All these lead into complications.
E20	B176	Repeat consider path (1.BxN'/5)	Now let's see, let's try once again.
	B177	1...B'xB	If he plays Bishop takes Knight, Bishop takes Bishop,
	B178	2.N-R4	then we can play Knight to Rook 4
	B179	B178 => N attack Q' and N defend NP	attacking the Queen and defending our Pawn at Knight 2 simultaneously,

	B180	3.N-B5 next White move after B179; possible	with the possibility of moving next move into Bishop 5
	B181*	B180 => N attack Black	putting a little more pressure on the - on Black
	B182*	B180 => 3...B'xN; follow B180	and perhaps persuading his Bishop to take the Knight at Bishop 5.
E21	B183	2.N-R4; repeat B178	If we play Knight to Rook 4,
	B184	Find 2...Q'-move	where can the Queen go
	B185	1.BxN'/5	if we play Bishop takes Knight -
	B186	1...B'xB; response to B185	to answer that Bishop takes Bishop,
	B187	2.N-R4; repeat(B183)	White follows with Knight to Rook 4.
	B188	2...Q'-QN-file not possible for B184	The Queen can go nowhere on the Knight's file.
	B189	2...Q'-R4 possible for B184	It can, of course, move to Rook 4 -
	B190	2...Q'-B2[retreat] or 2...Q'-Q1[retreat]; for B184	can move to Rook 4 or it can retreat to Bishop 2 or Queen 1.
	B191	2...Q'-R4, follow B183	If it moves to Rook 4,
	B192	3.N-B5	then we can play Knight to Bishop 5,
	B193*	B192 => N threaten Q'	again threatening the Queen
	B194	B193 => Q'-move[back] necessary	forcing it to move back,
	B195	3...Q-N3; for B194	probably again - probably moving back to Knight 3
	B196	B195 not desirable	and this Black would not do.
	B197	B195 => Black lose 2 tempo; => B196	He's lost two moves.
	B198	N'(B5) desirable for White	Our Bishop at Bishop 5 is in a good position,

	B199	B196 and B198 => not B186	so therefore he will not take the Bishop with the Bishop.
E22	B200	1.BxN'/5	Again we play Bishop takes Knight
	B201	1...P'xB response to B200	answered by Pawn takes Bishop.
	B202	B201 => P' isolated; not desirable for Black	This isolates a Pawn - its a tactical disadvantage.
	B203	B202 => not B201	It's doubtful that he'd do this.
	B204	B205 and B206 => B203	Besides
	B205*	2Pc bear on P'(Q4) possible	we can put two pieces on that Pawn right away
	B206*	B205 => Black defend P'(Q4) difficulty	and it would become hard to defend later on
E23	B207	(B199 and B203) => 1...N'xB response to 1.BxN'/5	so he will answer Bishop takes Knight with Knight takes Knight - with Knight takes Bishop.
	B208	Confirm B207	We'd answer it with Knight takes Bishop.
	B209	2.NxN'; follow move(B207)	Then if we play Knight takes Knight
	B210	B209 => N threaten Q'	the Queen is threatened
	B211	B210 => Q'-move necessary	and must move.
	B212		Well, therefore,
	B213	Move(B207) not possible	no he cannot answer it with Knight takes Knight
	B214	Confirm B210 and B211	because if he does play Knight takes Knight, the Queen is threatened and must move no matter where it moves.
	B215	(3.BxB' or 3.NxB'ch); follow move(B211)	We can either play Bishop takes Bishop or Knight takes Bishop check;
	B216*	3.NxB'ch desirable	Knight takes Bishop check is better
	B217	B216 => N fork R' and K'	being at fork with the Rook,

	B218*	B217 => Black lose Pc'	therefore he'd lose at least a piece.
	B219	Confirm not B207	All right, so he cannot play Knight takes Bishop if we play Bishop takes Knight
E24	B220	B219 => 1...P'xB necessary; follow 1.BxN'/5	Therefore - therefore he must play Pawn takes - Pawn takes Bishop
	B221	Find result(B220)	If he plays Pawn takes Bishop - what have we gained?
	B222	Remove 1(P bear on KB5)	We have gained - we have taken away one of the pieces - one of the Pawns on Bishop 5 square
	B223*	B222 => K'side attack with P(Kside) possible	thus making a Pawn push more reasonable.
	B224	2.P-B4 after B220 possible for B223	We can play Pawn to Bishop 4,
	B225	3.P-B5 next White move after B224; for B223	followed by Pawn to Bishop 5.
	B226	B225 => 1...P'xP necessary	This will,
	B227	Not B226	well, it won't force Pawn takes Pawn.
	B228	Not move(B226) => 4.P-KR4 next White move after B225 possible	However, we can if he does not take the Pawn, we can push on the other side - Pawn to King Rook
	B229	5.P-KR5 next White move after move(B228)	followed by Pawn to King Rook 5.
	B230	B229 => destroy structure (Kside)	This would mess up his King side
	B231	B230 => attack K' possible	and leave him open to an attack
	B232	B231 => checkmate(Black)	which should lead to an easy win.
E25	B233	Choose 1.BxN'/5	All right, so the best move is then Bishop takes Knight.
	B234	1...N'xB; response to B233	If it's answered with Knight takes Bishop

B235	2.N'xN	we play Knight takes Knight.
B236	1...P'xB response to B233	If it's answered with Pawn takes Bishop,
B237	2.P-KB4	we will play Pawn to Bishop 4.
B238	1...B'xB; response to B233	If it's answered by Bishop takes Bishop,
B239	2.N-R4	we play Knight to Rook 4
B240	3.N-B5 next White move after B239	and follow that up with Knight to Rook 5.
B241		Okay.

Notes for coding

B2 Refers to pieces and not men in light of B3.

B12 N(B3) rather than B(N2) in light of B16.

B16 "Attack" is probably better than "bear on," considering subject's quite consistent use of "threaten," "attack," and "bear on."

B50 Although "forgotten" implies a reference to B27 and B28, it is possible that the subject already sees 3.RxR'.

B54 The subject underestimates; in this position he can win a piece.

B87 "Bishop's Pawn takes Pawn" means BP'xN, since there is no Pawn capture on the board.

B88 The subject is not explicit, but the only obvious way to bring the Rooks into play is by doubling them on the QB-file; the reference to "two moves" in B89 supports this interpretation.

B90 We ignore the time dimension.

B104 "Knight 2" is "Rook 2."

B138 1.NxN' is possible rather than 2.NxN', which would imply a shift in B139, B140, etc. However, the comments in B145 and B146 support the choice of 2.NxN'.

B139 Possibly 2.NxN' is never considered and B138 is just a falter prior to B139.

B147 "Queen 2" means "King 2."

B148 See B147 note.

B168 1.N-K4 must be assumed, since otherwise the N defends B(R2). Also "Knight -" must mean NP, since there is no Q'xN move.

B175 Apparently subject is summarizing the entire prior analysis and not making a new judgment at this point.

B181 "Pressure" is not adequately rendered.

B182 "Persuade" implies that it is desirable for White to have N'xB, presumably to remove the B as a defender of the N(B3).

B193 N(B5) does not threaten Q'. If B192 were N(K5)-B4 then N(B4) would threaten Q'; but this seems most improbable in the light of subsequent behavior (e.g., B195).

B205 We ignore the time dimension.

B206 We ignore the time dimension.

B216 We ignore the comparison.

B218 We ignore the implication that there might be more than one piece lost.

B223 We ignore the comparison.