01 May 1986

Data Structure for the Use of Patterns in the Perceptual Ordering of the Game of Chess

Russ L. Hanna

Arlan R. DeKock
Missouri University of Science and Technology

Recommended Citation

Hanna, Russ L. and DeKock, Arlan R., "Data Structure for the Use of Patterns in the Perceptual Ordering of the Game of Chess" (1986). *Computer Science Technical Reports*. 8.
https://scholarsmine.mst.edu/comsci_techreports/8

A DATA STRUCTURE FOR THE USE OF PATTERNS IN THE PERCEPTUAL ORDERING OF THE GAME OF CHESS

R. L. Hanna* and A. R. DeKock

CSc-86-2

Department of Computer Science
University of Missouri-Rolla
Rolla, Missouri 65401 (314) 341-4491

*This report is substantially the M.S. thesis of the first author, completed May, 1986.

ABSTRACT

It is well known that the way one perceives a problem can influence the difficulty of solving the problem in a profound way. In the case of computer chess playing programs, one finds that most programs perceive the game in much the same way. They are all based on Shannon's original proposal for chess playing programs. His approach was to generate all of the possible combinations of moves up to a certain number of plays and then a subset of all combinations to a deeper level thereafter. Each of these moves would then be evaluated as to its relative worth. This paper lays the foundation for research in an alternate method of approaching the game based on how human experts perceive the board initially. A suitable data structure for this is then proposed and discussed.

TABLE OF CONTENTS

Abstract. ii

Acknowledgements. iii

List of Figures v

Chapter I: Introduction. 1

Chapter II: Review of Literature 4

Section A: de Groot's work. 4

Section B: The MATER Program. 8

Section C: The Perceiver Program. 9

Section D: The Patterns Discovered. 12

Section E: The MAPP Program 19

Chapter III: Discussion. 23

Section A: Background 23

Section B: The Data Structure 24

Section C: An Example 27

Chapter IV: Conclusion and Suggestion for Further Work . 38

Bibliography. 42

VITA. 44

LIST OF FIGURES

Page

Figure 1 Example of the EPAM net. 21

Figure 2 General scheme for pattern storage 25

Figure 3 Examples of patterns in the same class 28

Figure 4 A numbering scheme to specify the placement of pieces. 29

Figure 5 A scheme for relative placement of pieces. . . 31

Figure 6 Two different pattern categories 32

Figure 7 Specification for relative placement of a Knight 33

Figure 8 General pattern description of figure 3. . . . 36

CHAPTER I

INTRODUCTION

Since Shannon proposed a method for a computer to play chess, there has been a huge amount of work done in an effort to produce a computer which could seriously challenge a human opponent. After 37 years we are only now beginning to see chess programs challenge a human expert at the game. It is obvious that writing a good chess program is a difficult task, but it may be that part of the difficulty rests with the particular method chosen to solve the problem. This method is influenced by the way the game is being perceived by the programmers of today which is different from the way a human chess expert perceives the situation. It has been put forth that further significant improvement in programs will probably not come from a continuation of current methods (3, p. 35).

The key to understanding chess is to understand the perceptual processes involved in playing the game (4, p. 56). In fact, it has been said that a game of chess is not mastered by the highly intellectual, but rather by the highly perceptive (3, p. 51). The way a problem is perceived can have an enormous influence on the difficulty of solving it. To illustrate the importance of perception in problem solving consider the following problem proposed by Posner.

Two train stations are fifty miles apart. At 2 P.M. one Saturday afternoon two trains start toward each other, one from each station. Just as the trains pull out of the station, a bird springs into the air in front of the first train. When the bird reaches the second train, it turns back and flies toward the first train. The bird continues to do this until the trains meet. If both trains travel at the rate of twenty-five miles per hour, and the bird flies at one hundred miles per hour, how many miles will the bird have flown before the trains meet? (11)

Upon hearing this problem many people will try to solve it in terms of the bird's flight pattern. They will attempt to find out how far the bird flies in each direction and then add these separate figures to obtain the answer. This can be a nearly impossible task. The problem becomes much easier if it is simply noticed that the trains will meet in one hour. The bird is flying one hundred miles per hour for the duration of this hour, therefore the bird flies a total of one hundred miles. The task became much easier, not because the problem itself changed, but because the perception of the problem changed.

Chess is particularly well suited for study on this point, since it provides a way to study and compare two different perceptions of a problem. There are at least two distinct approaches to playing the game: The human chess expert's approach and the traditional game tree approach used by chess playing programs. These two approaches stem from two different perceptions of the board. In the human method, the perceptual phase is thought to be the "really significant part" of the problem solving process (11, p.

473). If the human perceptual phase in chess can be understood then this will not only indicate a method of how to play the game better, but will, ideally, also teach something about the role perception plays in general problem solving.

CHAPTER II

REVIEW OF LITERATURE

Chess has long been an object of study by many scientists. In fact, serious research into this problem stretch back into the late 1800's. These early researchers accepted many of the myths of the time concerning the methods employed by players who had mastered the game of chess. Many of these assumptions were in error which, in turn, limited the usefulness of the research. These myths were later dispelled by the first study discussed below.

SECTION A: DE GROOT'S WORK

The first and--to this date--the most comprehensive work regarding the study of the abilities and methods that make some people better chess players than others was done by A. D. de Groot and his colleagues in Holland in 1965 (7). De Groot studies several subjects whose playing level ranged from grandmaster to class C on the USCF rating scale. At first, de Groot tried to determine differences among players by examining verbal protocols of all the subjects. A verbal protocol refers to the statements made by a person who is "thinking out loud" while solving a problem. In this study, each player was asked to analyze a position with the intent of finding the best move. These positions were taken from actual chess games between master level players which had been played recently. As the subjects proceeded with the analysis, they were asked to think aloud so that an indication of their thought processes could be recorded (verbal protocols are considered only an approximation of the actual cognitive functioning). The intention was to study these protocols in an attempt to find any differences in the cognitive processes between the various levels of players. Up to this point, de Groot was operating under the assumption that the stronger players had superior thought processes to those of the weaker players (i.e. better for chess) and a study of these processes and methods should, therefore, reveal the reason why some players are better than others.

The results surprised de Groot. He found that there was no clear difference between the players on any measure except one: stronger players gave more accurate estimates of the values of the moves which they chose as a result of their analyses of the various positions than did the weaker players. However, despite the fact that good players gave a better estimate of the move than did the weaker players, all the subjects used the same methods and processes to arrive at their respective decisions. It was also noted that good players spent most of their time investigating promising options while weaker players spent much of their time looking down blind alleys. It seemed that there was something about those particular positions explored which attracted the attention of the master players before they began their analysis. In other words, stronger players seemed to know in advance which two or three moves were the best and then proceeded to study only those moves in an effort to find the best option among them.

It was also found that many earlier beliefs regarding strong players were false. Before, it had been assumed that strong players searched many more plys than weaker players, but it was here discovered that grandmasters rarely searched more deeply than did a class C player. In fact, they often searched less deeply. Also, the number of moves actually explored dropped to the range of 20-76 rather than the "hundreds" previously thought to be examined. This range is also about the same as for any other level of player (p. 38).

With these results in mind, de Groot postulated that the stronger players were perceiving the positions on the board differently than the weaker players. It should be emphasized that this perceptual phase is prior to the problem solving phase. The perceptual phase is characterized by a period of gathering information about the problem in preparation for the problem solving phase. The problem solving phase is the time when the problem space is actually searched for a solution. De Groot explored this hypothesis with another experiment.

In this new task the players were asked to look at a chess board which had been set up with a "quiet" position taken from a recent game between master chess players. The subjects were allowed to look at the position for only a short time (on the order of 5 seconds or less) and then asked to reconstruct from memory the board they had just seen. Weaker players were allowed to use a chess board to set up the pieces. Stronger players simply called out the pieces along with their positions as they remembered them.

The results of this experiment were also surprising. Strong players (master or grandmaster) would get 23 or so out of 25 pieces correct for an average of approximately 93%. Experts recalled approximately 72%, and the weaker players averaged 51%.

De Groot felt that this result was much more than just an interesting side effect of staring at chess boards for years. He thought that this was somehow central to chess skill.

It was then postulated that good chess players perceived the board not as a collection of pieces, but rather as a collection of "chunks". These chunks were easily recognized and remembered by the strong players who had seen them many times before in games they had played or in the course of study. What a chunk looked like and what bound it together would be clarified by later research.

The following points contain a brief summary of the important findings of de Groot which formed the basis for almost all of the work to follow:

1) The general thought processes and methods used are the same for both strong and weak players.
2) Myths about how experts play chess were dispelled. Among these were the myths that master chess players search more plys (deeper) than weaker players

players and the idea that strong players search a larger portion of the problem space than do weaker players.

3) Point two indicated that strong players must perceive the board differently than weaker players.
4) Master players are able to reproduce a board almost perfectly after only a few seconds of exposure.
5) The existence of perceptual chunks was postulated.

SECTION B: THE MATER PROGRAM

In 1966, George W. Baylor and Herbert A. Simon (1) attempted an early simulation of some of the general perceptual processes which players use during the problem solving phase while playing the game of chess. This program ignored the perceptual chunks, since little was known about them at the time, and was restricted to a rather small domain, but it did incorporate some of the important methods, which were uncovered by de Groot earlier, used by chess players to limit the search space. There were two versions of the program called MATER I and MATER II. Their abilities are best summarized by Baylor and Simon (1, p. 441).

> MATER I solves combinations which consist of uninterrupted series of checking moves, given that the defender at no node in the verification tree has more than four legal replies; MATER II solves combinations that begin either with checks or with one-move mate threats and checking moves thereafter.

Even though the programs are rather restricted in the types of problems they can solve, they are still an impressive display of what can be accomplished using an approach which tries to take advantage of human methods of heuristic search rather than the usual game tree approach.

SECTION C: THE PERCEIVER PROGRAM

In 1969 Herbert A. Simon and Michael Barenfeld (12) attempted to explain the initial perceptual phases of problem solving in terms of an information processing model. This model of the perceptual phase of chess was incorporated into a program called Perceiver. Perceiver was based on a significant amount of earlier work which will be summarized first.

Early work on perception was done by studying the eye movements of subjects by using apparatus which could determine and record where the subject was looking at various times throughout the perceptual phase (1, 8, 10). Chess was chosen as one of the problems to study because a great deal was already known about the game.

Subjects were set up with the above mentioned apparatus and then given a chess position to analyze with instructions to find the best move. The apparatus then determined and recorded where on the board each successive fixation fell with an accuracy of approximately one square. With this information in hand, it was then possible to trace through the various fixations in order to plot and study how the subject was scanning the board in the initial perceptual phase.

It is, of course, impossible to know exactly what takes place in the mind of the subject as he is scanning the board which means that it is impossible to be sure exactly what information is being transferred from the board at each fixation. Taking this into account, the data was analyzed and seemed to indicate that the subjects were acquiring information about the piece or pieces at or near the point of fixation. The data also indicated that information was being gathered about pieces by the subject's peripheral vision around the fixation point which had a significant chess relation (attack, defense, proximity. . .) to a piece at the fixation point. Also, the information gathered in the peripheral vision seemed to be used to establish where the next fixation point was located. The board could then be viewed as a net of relations which guide or pull the eye to the important locations.

Perceiver attempted to organize all the perceptual processes already contained in MATER into a new program which would also simulate those eye movements mentioned above. There were four different chess relations noticed by the Perceiver program: 1) pieces that defend the fixated piece 2) pieces that attack the fixated piece 3) pieces that are defended by the fixated piece and 4) pieces that are attacked by the fixated piece.

The Perceiver program itself was made up of two parts. The first part scanned the board in the manner mentioned above in an effort to find the pieces which had the best chance of being involved in the good moves. The second part used perceptual processes to search for a move, as in MATER.

In the running of the first part of Perceiver, it was found that the computer "eye" was continually drawn to the area of the board which contained the greatest density of relations (12, p. 479). It was determined that, while doing this, Perceiver traced out a path which was not dissimilar in any way from that of a human. More importantly, Perceiver showed an "almost complete preoccupation with the ten critical pieces" (12, p. 477). The ten critical pieces were central in the understanding of the position and also included the pieces on the board which need to be studied to find the best move. This preoccupation with a few pieces is similar to human play and seems to indicate that Perceiver had captured something of the essence of the perceptual process which allows a strong player to see the good moves before he begins to analyze the position.

The second part of Perceiver uses the same processes to search for a move. In the example given in the paper (12, p. 479), the critical move is indeed discovered by scanning the board in a manner similar to that of a human. Again, this indicates that the processes embodied in the Perceiver program had captured something of the essence of the human method contained in the problem solving phase.

At the end of this paper is found the first estimate of the number of chunks or patterns that the master level chess player probably needs in order to play at that level. At this time no one really knew the exact nature of a pattern so an assumption was made about their probable nature. It was conjectured that the patterns in chess were comparable to visual word recognition vocabularies of people learning to read the English language (12, p. 481). After a brief analysis a range of 10,000 to 100,000 is quoted as the probable range for the number of patterns that the master chess player needs, with 50,000 being a likely best figure. This figure was revised by later research.

SECTION D: THE PATTERNS DISCOVERED

In 1973 William G. Chase and Herbert A. Simon (4) replicated and extended the earlier work done by de Groot on perception (the recall task) in an effort to "discover and characterize" the perceptual chunks that had only been postulated up to that point (4, p. 56). This is the first work that actually tried to find out what a chunk looked like and what chess relations held it together. The first part of the experiment was concerned with replicating and confirming de Groot's findings. In Chase and Simon's experiment, however, two important elements were added to strengthen the experimental method. The first was a valuable control element which had been completely missing in de Groot's experiment. This control feature consisted of the addition of several recall trials for all subjects on randomized chess patterns. This was added in an effort to determine whether the stronger players' higher performance on the recall task was due simply to superior memory ability. The second important addition was the inclusion of a beginner as one of the subjects to be tested. The testing of a beginner allowed Chase and Simon to see how someone would perform on this task before he had been "trained" by playing experience to perceive the chessboard in any special way. Also, viewing all the subjects together may reveal any progression of method a chess player goes through as he becomes more proficient at the game. The other two subjects were a master and a class A player.

Chase and Simon's reconstruction task confirmed de Groot's earlier work. The tendency of stronger players to score better was again observed (81% for master, 49% for class A, and 33% for beginner). The slightly lower percentages seen here can be explained by considering the differences in the stimuli between de Groot's recall task and this one. The important question at this point was this: Is the ability to reconstruct a board almost perfectly a matter of chess specific memory capacity or does the chess master possess some natural superior mental ability that the weaker player is lacking? This question was answered by testing recall performance on a randomized board position (i.e. the control element mentioned above). In this task the three subjects were shown randomized board positions and asked to reconstruct the board from memory in the same manner as before. In this task all the subjects performed roughly equally as poorly. In fact, the master performed slightly worse than the beginner. This showed that the ultra-high performance by the master on the recall task was due to chess specific memory and not to some superior mental ability possessed by the master.

The second part of the paper is concerned with the discovery of the perceptual patterns themselves. To help remove any artifacts in the data which might be introduced by the method and to reinforce any findings, Chase and Simon used two completely different experiments to try to uncover the nature of chunks. The first experiment was termed the perception task and the second was termed the memory task.

In the perception task two boards were provided for the subject, one to his left and one directly in front of him. A partition was placed between the two boards such that the subject was unable to see the board on the left, on which a chess position was set up. The partition was then removed and the subject was asked to reconstruct the chess position to his left on the board in front of him as quickly as possible, looking at the chess position as often as he wished. This entire operation was recorded on videotape.

In the memory task the boards and the chess positions were set up in the same manner as in the perceptual task. However, when the partition was removed, the subject was only allowed to view the position for 5 seconds before the partition was replaced. The subject was then asked to reconstruct the position from memory by setting up the pieces on the board in front of him, as in the perceptual task. If the subject failed to perfectly reconstruct the position, the board in front of him was cleared and the task was repeated with the same position. This procedure was repeated as many times as necessary to achieve perfection with the given position. This process was also recorded on videotape.

Both of the above tasks depended on earlier findings concerning the time required to transfer information from long-term memory into short-term memory. Dansereau (6) found that approximately 2 seconds were required to begin processing a chunk whose label was being held in short-term memory and approximately 300 msec were needed to transfer each successive element of the chunk into short-term memory once the processing had begun. The assumption was that the master players had a label (name) for the chess positions they recognized. Then, on the recall task, rather than storing the entire board in memory as individual pieces, the master simply stored the labels representing those patterns in short-term memory and reconstructed the board from this information.

These findings were used to delineate the chunks on the board by pinpointing chunk boundaries. In the perception task there were two time intervals to analyze. The first was the within-glance interval which corresponded to the time interval between pieces placed without looking back at the original position. The second was the between-glance interval which corresponded to the time interval between two pieces separated by a glance back at the original position.

The within-glance times averaged 1.22 seconds for the master with only one time exceeding 2 seconds, .99 for the class A player, and .89 for the beginner. For the between-glance times, the averages were 2.86, 3.3, and 3.58 seconds for the master, class A, and beginner respectively. Referring to the above statements about processing time in short-term memory, it can be assumed that the within-glance times correspond to the transfer of successive items of a chunk into short-term memory since the times are too short for the processing of new patterns. Similarly, the between-glance times, being over 2 seconds, should correspond to chunk boundaries i.e. a new chunk is being processed. One can then conclude that the pieces that were placed after each glance should correspond to a perceptual chunk.

In the memory task there were no times corresponding to the glance times in the perception task to analyze. Instead the hesitations which were longer than 2 seconds were looked for and noted. It was assumed that these hesitations corresponded to chunk boundaries, and the resulting chunks were then compared to the chunks found in the first task to see if the frequency distributions were similar. If they were found to be similar, this would indicate that the same types of chunks had been found. Upon comparison it was discovered that the patterns found in the two tasks did, in fact, correlate very closely with one another.

The next question to be answered was that of just what characterized a pattern. To answer this question five different chess patterns were chosen for study. These relations were labeled attack (A), defense (D), same color (C), same piece (S), and proximity (P). These categories were also combined in order to form all of the possible combinations of relations. For example, a DPS refers to a piece that is defended by, proximite to, and the same piece (type) as some other piece on the board. The patterns found in the two tasks were then analyzed to determine whether there were any relations which had a higher or lower frequency of occurance than would be expected from a random placement. When the within-glance times from the perception task and the hesitations shorter than 2 seconds from the memory task were analyzed the results showed that several relation combinations had a significantly higher probability than would be found by chance. These relation combinations include AP, DC, DPC, PCS, and DPCS. There were also three combinations which had lower than chance probabilities. These were C, S, and the null relation.

This does not mean that such relation combinations are automatically chunked when they are discovered, but rather that they have a good chance to be chunked. Similarly, the discovery of the null relation between two pieces indicates that the two pieces will probably not be chunked, although they may be in some circumstances. The processes which actually determine when these pieces are chunked and when they are not have yet to be completely uncovered. It is also important to note that the data is similar for all the subjects; even the beginner (4, p. 65). This reinforces the belief that the master player has not found some "better" way to organize the board than the beginner. He merely has more practice at the task.

Analysis of the between-glance times from the perception task and the hesitations longer than 2 seconds from the memory task indicated that these pieces were placed with a probability very close to chance. This allows the conclusion that the pieces very likely had no connection in the subject's mind, and therefore, correspond to a chunk boundary.

Looking over the data it is possible to roughly determine the size of the pattern the chess player uses. The average chunk size for the master turned out to be 2.2 pieces with a maximum, in this test, of 7 and a minimum of 1.

The authors also point out that it is highly probable that the master uses these chunks to encode information about the position itself. When the patterns are recognized this information becomes activated and suggests strategies to the player. This is how the chess patterns are linked to chess skill (see also 3, p. 49).

Finally, we have a new estimate of the number of patterns a chess master knows. No precise figure is given, but the previous statement by Simon and Barenfeld of 50,000 is now thought to be too large. The master's performance can be accounted for if we postulate a long-term memory capable of recognizing a few basic patterns and their variations. These general patterns are characterized by the authors in (4, p. 80):

1) A variety of chunks consisting of Pawns (and possibly Rook and minor pieces) in common castled-King configurations;
2) A variety of chunks consisting of common first-rank configurations;
3) A variety of chunks consisting of common Pawn chain, Rook pair, and Rook and Queen configurations;
4) A variety of common configurations of attacking pieces, especially along a file, diagonal, or around an opponent's castled-King position.

SECTION E: THE MAPP PROGRAM

In 1973 Herbert A. Simon and Kevin Gilmartin (13) implemented a program to simulate the memory recall task in chess, using all that was known about human performance in this particular task. To do this they combined the Perceiver program with the EPAM discrimination net. The EPAM net embodies a theory of human long-term memory; these two components together simulated the eye movements and the long-term memory of humans in the memory recall of chess positions task. A short-term memory component was added to complete the model which was named MAPP (Memory Aided Pattern Perceiver).

The program had four logical parts, the first of which was a pattern learner. This part allowed the EPAM net to be grown (by saving patterns) by reviewing some of the chess literature in preparation for the memory recall task. Various sizes of nets were grown and tested in the recall task. The larger nets--the ones containing the most patterns--showed the best performances.

The second part of the program was the EPAM net itself. This net consisted of a simple binary tree which specified the pieces along with their rank and file on the board as in figure 1. It should be noted that this representation is somewhat limited, since it provides no ability to substitute a new piece into an existing pattern at any point in the pattern where the piece type may be variable. Also, the placement of pieces is restricted in that a piece can only be on the one square of the board specified by the pattern. If the very same pattern were to be shifted even one square in any direction, it would not be recognized unless this pattern had been stored redundantly for every possible position.

The program's third part was the salient piece detector. This part consisted of a modified version of the Perceiver program, which identified any "interesting" pieces on the board. The "interestingness" of a piece was a function of the type, proximity, and color.

Once the salient piece had been found, the fourth part of the program took over: the pattern discriminator. This part would search through the EPAM net for a pattern

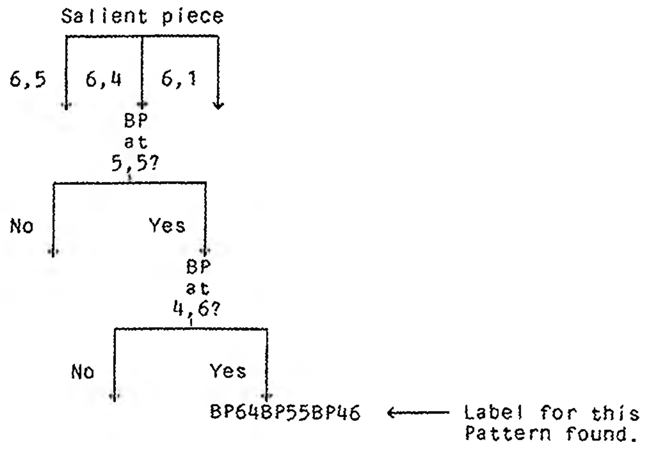


Figure 1. Example of the EPAM net.

starting with the salient piece.

This program proved to be a good simulation of human performance in the recall task. The pattern sizes averaged 2.45 pieces as compared to 2.2 discovered for the master in the previous study, and averaged about 73% in the recall task when the largest EPAM net was used. This program also provides a better estimate of the number of patterns needed by the master player; by studying MAPP and evaluating the number of patterns it needed to reconstruct the board at various levels of proficiency. After a moderately lengthy analysis, Simon and Gilmartin arrive at a figure of about 10,000 with an upper bound of 100,000. It should be noted here that this figure is based in part on the EPAM net's method of storing the patterns. The particular method employed did not take into account any of the general patterns mentioned before. If this were considered, the figure of 10,000 would probably drop by some significant amount.

CHAPTER III

DISCUSSION

SECTION A: BACKGROUND

It is not the purpose of this paper to specify the details of a program which would play chess using chess patterns and the later proposed data structure, but a brief overview of the general way in which such a program would most probably function is in order, to help explain the form of the data structure given.

Since it is the aim, eventually, to emulate the human perceptual breakup of the chess board, it is necessary to simulate two processes. The first is the propensity of the human eye to be drawn to the area with a high density of relations on the board. This simulation would be accomplished by scanning the board and assigning a number to each piece indicating how "interesting" that piece appears. This is the same manner in which the salient piece detector in MAPP assigned a saliency score to each of the pieces of a position. This notion of interestingness would consist mainly of, but not necessarily be restricted to, the density of the relations emanating from that piece. The processes involved in this evaluation are, at the present, somewhat unclear, but its probable nature can be conceived in a general way. The different combinations alluded to by Chase and Simon (4) would be given different weightings based, in part, on their relative probabilities of forming a tie between two pieces. The sum of these weights would be added up to form the main part of the number indicating the interestingness of the piece. The pieces with the highest number would then correspond to the points on the board where the computer eye should be drawn. Having assigned a number to each piece, the first part of the simulation is complete.

The second part of the simulation would actually involve searching out the patterns themselves. In humans this seems to be done by searching in the peripheral vision along the lines of the relations emanating from a piece, as discussed before, and noting any ties that may be formed to other pieces. Also, recall that this peripheral searching indicates to the human player where the next fixation point should be located. This is one point where the program would probably deviate slightly from the human player. The program needs only to select the piece which is the most interesting at any time. Searching around this piece first would give the best chance for the program to find the biggest pattern on the board at that time, thus eliminating the maximum number of pieces from further consideration. This phase is accomplished with the data structure shown in figure 2.

SECTION B: THE DATA STRUCTURE

At the top left of fig. 2 is the interesting piece noted before. The interesting piece will always be the beginning point for the search. There would be a maximum of twelve of these points; one for each type of piece on the

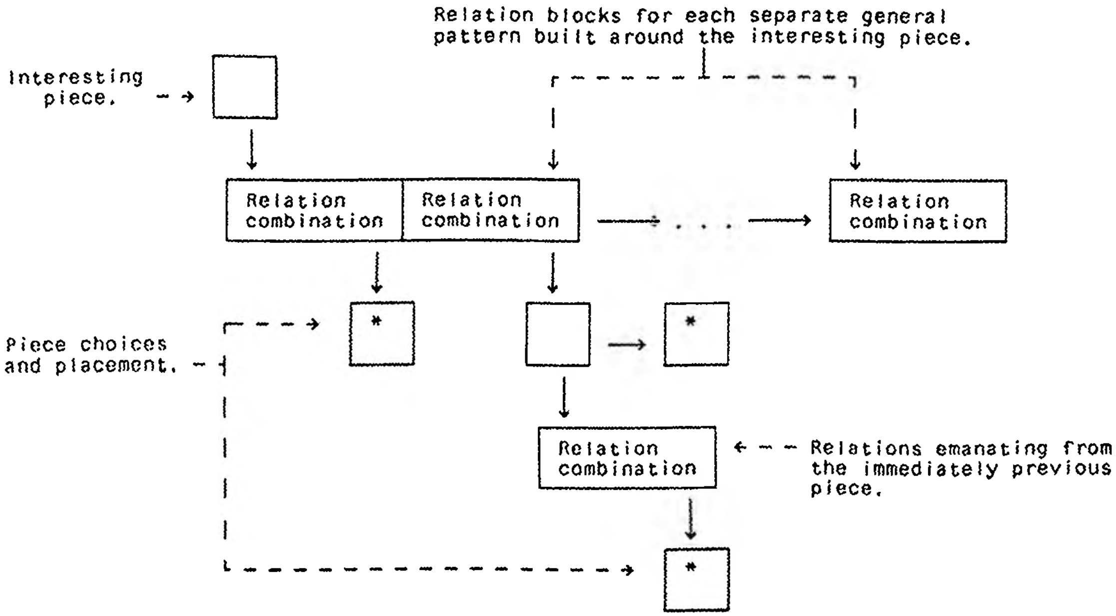


Figure 2. General scheme for pattern storage.

board.

Directly below the interesting piece is a list of relations. Each group of relations in this list contains the relation combinations that emanate from the interesting piece for exactly one pattern. In other words, each block of relations below the salient piece will contain the relations that would be searched by the peripheral vision of a human player. When a new general pattern is discovered around this piece its relation block is simply added to the end of this list.

For each of the relation combinations within the blocks shown, there is a list containing the pieces which are allowed to be the object of each particular relation, along with the allowed locations for these pieces. If the program has traced far enough along a relation string to find part of a pattern, then the block is marked as such by an asterisk. If the chain of relations continues on past this point, then that piece is given a relation list identical to that of the interesting piece, and the search is continued in exactly the same way as before.

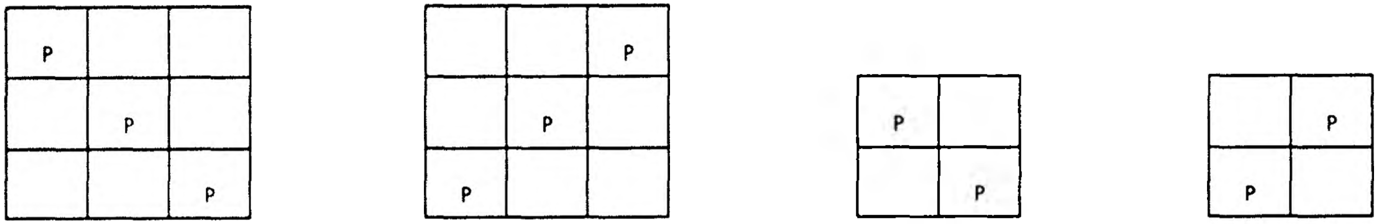
Each pattern will also have associated with it a set of recommendations for further play beyond this point in the game. After a pattern has been discovered, its proposal is activated and passed to the program. These suggestions will be of the same nature as those activated in a human player when he recognizes a pattern. The program will take these plans of attack from each pattern and choose a limited number of goals, which would then be explored. These plans will be activated by the retrieval of the pattern itself and are not actually a part of the data structure shown.

SECTION C: AN EXAMPLE

The functioning of the data structure will be clarified with an example. Suppose the pattern shown in figure 3a is given. The program has decided that figure 3a is equivalent to figure 3b, c, and d. All of these patterns need to be represented by one general pattern description.

The first problem to be solved is how to describe this pattern for any location on the board. Figure 4 shows the logical view of the board employed to solve this problem. The squares of the board are numbered sequentially from 1 to 64 starting at the upper left corner as viewed from the side whose pieces are being analyzed. This means that the square called number 1 by the white player is called number 64 by the black player. Piece locations can be specified by simply referring to the square number of that piece as viewed from the side of the board to whom that piece belongs. The solution is to leave the specific square number of the interesting piece out of the description of the pattern. Then, when the patterns are being searched, the interesting piece in the pattern description will be translated to the actual board position of the piece being examined.

The above solution raises a second problem. If the actual location of the interesting piece is not known



a. b. c. d.

Figure 3. Examples of patterns in the same class.

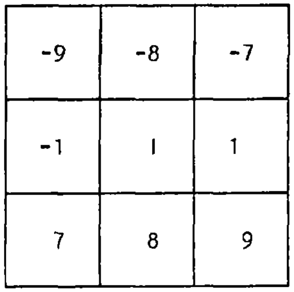
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 4. A numbering scheme to specify the placement of pieces.

beforehand then the location of the other pieces in the pattern needs to be specified in terms of each other rather than as absolute positions on the board. The relative placement problem is solved by the scheme shown in figure 5. The point on this section of the board where the interesting piece is located is indicated by the "I". The number of the square on which I is sitting, as reckoned by the above numbering, will be denoted by z. The numbers appearing around the location of the I are the amounts that must be added to z to obtain the number of the particular square where the number appears. For example, call the square number of the lower right Pawn shown in figure 3c x. The other Pawn shown would be on the square numbered x-9.

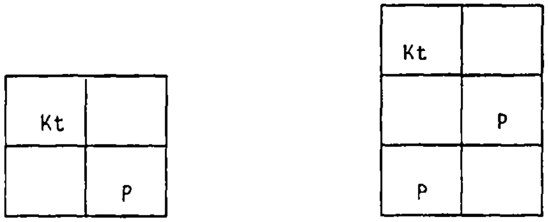
Now a method using the above systems can be specified to represent placement of pieces in relation to each other. This method will also specify how restrictions can be placed on certain pieces if so desired.

Consider figure 6. The goal is to specify that the Knight can be placed above and to the left of a Pawn as shown in part a. There is only one circumstance in which this placement will be considered a change in the pattern. That situation is shown in part b. The part of the data structure dealing with this situation would then be specified as shown in figure 7. This expression has two parts. Part one specifies which pieces are allowed as the object of the relation being considered at this point. In the above example, a Kt appears which indicates that only a



-9	-8	-7
-1	I	1
7	8	9

Figure 5. A scheme for relative placement of pieces.



a. b.

Figure 6. Two different pattern categories.

Kt	-9(C) AND NOT 7(P)

Figure 7. Specification for relative placement of a Knight

Knight is allowed in this position. Part two of this expression is a Boolean expression indicating the options for placement of the piece(s) specified in part one. Part two also contains any restrictions. The letter placed in parentheses after the number indicates the piece or type of piece that the number refers to in this expression. The letters will have the following meanings:

C - The piece(s) referred to in the first part of the expression.

R - Any and all pieces.

P, R, Kt, B, K, Q - The standard chess symbols for the various pieces.

Other letters could be added to indicate specific combinations of pieces, but these will not be discussed here. The number denotes a displacement from a piece which is currently under consideration, so, the C in figure 7 refers to the piece or pieces that are being placed at this time. In this example the C refers to the Knight. Any other letter refers to a restriction of some kind. The P after the 7 means that this displacement refers to a Pawn. The representation in figure 7 can then be read as follows: The Knight may be placed at a displacement of -9 from the current location when there is not a Pawn at displacement -7. If a restriction is not specifically stated then it is assumed that none exist. The factors of this specification are compared to the arrangement of the board and are replaced with the value of true if the conditions are all satisfied. The conditions outlined in a specification such as this are satisfied if the type(s) of piece(s) is (are) correct and the location expression evaluates to true.

Using the above scheme, figure 8 shows a representation of the various Pawn chains presented in figure 3. The interesting piece is again located in the upper left of figure 8. In this case, the lower Pawn has been selected as the most interesting piece of these patterns. The interesting piece has the symbol P which stands for a Pawn. The x to the right of it is simply a placeholder and will be filled in with the actual location of the Pawn on the board which is under scrutiny.

Directly below the interesting piece is the relation combination along which the search for the next piece will be located. In this case, the DPCS is the only relation combination to consider. The letters pertaining to relation combinations used here have the same meaning as those discussed in the previous section.

Underneath the first DPCS combination are the piece choices which are allowed along this relation from the first Pawn. The only symbol here is a P, which means that only a Pawn is acceptable in this location. To the right are found the allowable locations for the Pawn. A displacement of either -9 or -7 from the first piece is indicated as being acceptable with no restrictions. Also, note the asterisk, which indicates that, at this point, a pattern has been defined. In particular, the patterns shown in figure 3c and

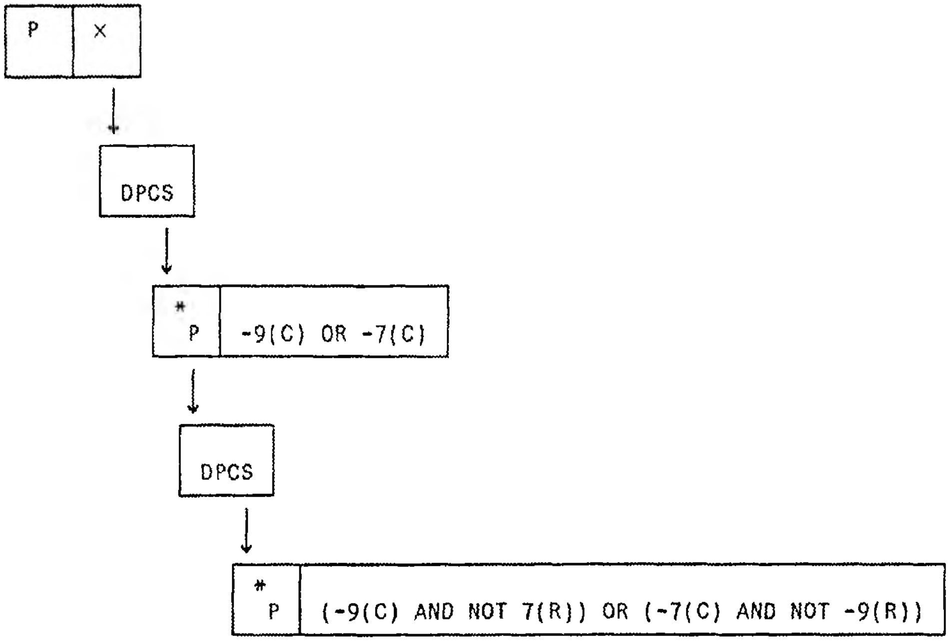


Figure 8. General Pattern description of figure 3.

d have been specified. From this point pieces are simply added on to this representation, along with any appropriate changes in restrictions, to obtain new and larger patterns that fit the general category type represented by this particular instantiation. This approach is typical of human play.

Below this second Pawn is another DPCS. This functions in exactly the same way as the DPCS relation above, except that the search is now along this relation from the second piece rather than the first.

There is another Pawn below the second Pawn. Again, it is the only symbol present which means that only a Pawn is acceptable in this location. To the right are the allowable locations for this Pawn. It is found that a displacement of -9 is acceptable as long as there is not a piece at a displacement of 7. Similarly, there may also be a Pawn at displacement -7, so long as there is no piece at displacement 9, as indicated by the second half of the specification.

This data structure can be thought of as a movable, flexible template which is positioned over the pieces in question to determine if there is a match between the pattern and the pieces on the board.

CHAPTER IV

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

It is clear that the human method of playing chess is complex and requires much more study in order to be able to write an effective program to play the game. Still, enough is known that it is reasonably certain that the data structure discussed herein is suitable for the task.

As mentioned before, the chess relations emanating from a piece are important in both the initial perceptual phase and the subsequent problem solving phase. This data structure is suitable because the patterns are stored in terms of their radiating relations rather than by spatial attributes as previous attempts have done.

This data structure can be employed in the two different ways depending on how the program functions. One way would be to have the program break the board up into patterns and then simply search the data for a match. With this approach the various heuristics and methods used for chess would be embodied in the program. This would also necessitate a separate mechanism to build and store the patterns before the program could use the data.

The other possibility would be to incorporate general heuristics in the program which could take advantage of past experience without requiring a change in the program. With this method the program would learn about the game by storing patterns and associated strategies based on what it encountered during a game. Then, instead of simply searching the data structure for a match, the program would use the patterns to guide and enhance its search around the board during the perceptual phase. This seems to be the more logical use for this data structure.

The second use of the data structure is more exciting because it leads to a more interesting conclusion. Visualize a program which would be a general strategist. This program would be able to play a number of tactical games once it was given the rules and the appropriate data. This type of general thinker is more desirable than a chess-specific program because it is possible to learn much more about a true general "thinking machine" from this model than the first one discussed.

Regardless of the approach taken, the program will still use the processes employed in the Perceiver and MATER programs to do the initial breakup of the board and to perform the subsequent search for the next move. This process was discussed at length on pages 23 and 24 of this paper.

The final problem remaining concerns how the suggestions for further work are going to be found and then associated with the patterns.

Pioneering work on this problem has already been done by Albert L. Zobrist and Frederic R. Carlson, Jr. at the University of Southern California (15). In their paper they describe a program developed at U.S.C. which is able to take advice from a human chess expert. What makes this program so interesting is how this knowledge is transmitted from the player to the program. The components of this process can be divided up into two parts. The first part is the language used to outline the advice to the computer and describe the situation in which the advice is to be used.

This language was invented just for this program. It is a simple language which enables the human player to describe general patterns of the same form discussed before. The advice is then appended to this pattern description and both components are submitted to the program.

The second component takes this pattern and advice and stores this for future reference. During a game, the program will search its patterns for a match and will act on any advice that it finds.

This program is useful because it gives a framework for encoding pattern related advice on chess without the need to change the code of the program. This, then, provides a way to associate general proposals with particular patterns. Also, this program is designed to accept generalized concepts of the nature generated by a human chess expert.

It can be seen that the addition of a pattern building ability and the routines which would allow the program to generate its own suggestions to this program would result in the first program to learn and then play by itself as a human would.

The above models immediately suggest where further work should be concentrated. First, the way humans acquire these patterns over the years is still unclear. All that is definitely known is that there seems to be a gradual shift from playing a mostly cognitive game to playing a mostly perceptive game. Before a program could be expected to perform this task, it is necessary to understand the human process.

Second, a study of just how the appropriate strategies are assigned to the patterns needs to be done. Again, it is not possible to program a computer to emulate the human at this unless the process is fully understood.

Once the two above processes are understood, it should be possible to use de Groot's findings in conjunction with them to write a complete program to emulate the entire process of human chess playing.

BIBLIOGRAPHY

1. Baylor, G. W. and H. A. Simon. "A Chess Mating Combinations Program." In AFIPS Conference Proceedings, 1966 Spring Joint Computer Conference, Washington, D.C., 1966.

2. Berliner, Hans and Murray Campbell. "Using Chunking to Solve Chess Pawn Endgames." Artificial Intelligence, 23 (1984), 97-120.

3. Charness, Neil. "Human Chess Skill." In Chess Skill in Man and Machine. Ed. Peter W. Frey. New York, Springer-Verlag, 1983, pp. 34-53.

4. Chase, William G. and Herbert A. Simon. "Perception in Chess." Cognitive Psychology, 4 (1973), 55-81.

5. Chase, W. G. and H. A. Simon. "The Mind's Eye in Chess." In 8th Annual Psychology Symposium Volume: Visual Information Processing. Ed. W. G. Chase. New York, Academic Press, 1973, pp. 215-281

6. Dansereau, D. An Information Processing Model of Mental Multiplication. Unpublished doctoral dissertation. Carnegie-Mellon University, 1969.

7. De Groot, A. D. Thought and Choice in Chess. The Hague: Mouton, 1965.

8. De Groot, A. D. "Perception and Memory Versus Thought: Some Old Ideas and Recent Findings." In Problem Solving. Ed. B. Kleinmuntz, New York: Wiley, 1966

9. Ellis, Henry C. and R. Reed Hunt. Fundamentals of Human Memory and Cognition. Wm. C. Brown Company Publishers: Dubuque, Iowa, 1983.

10. Jongman, R. W. Het oog van de meester. Amsterdam: van Gorcum, 1968.

11. Posner, M. I. Cognition: an Introduction. Scott, Foresman: Ill., 1973.

12. Simon, Herbert A. and Michael Barenfeld. "Information-processing Analysis of Perceptual Processes in Problem Solving." Psychology Review, 76, no. 5 (1969), 473-483.

13. Simon, Herbert A. and Kevin Gilmartin. "A Simulation of Memory for Chess Positions." Cognitive Psychology, 5 (1973), 29-46.

14. Tichomirov, O. K. and E. D. Poznyanskaya. "An Investigation of Visual Search as a Means of Analyzing Heuristics." Soviet Psychology, winter 1966-67, 5, 2-15 (trans. from Vopresy Psikholdodii, 1966, 2(4), 39-53).

15. Wilkins, David. "Using Patterns and Plans in Chess." Artificial Intelligence, 14 (1980), 165-203.

16. Zobrist, Albert L. and Frederic R. Carlson, Jr. "An Advice-Taking Chess Computer." Scientific American, 228 no. 6 (1973), 92-105.