GENERALIZED CHESS ENDGAMES

By

BRUCE ALLAN MOSER

A SENIOR RESEARCH PAPER PRESENTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE OF STETSON UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

STETSON UNIVERSITY

2003

ACKNOWLEDGMENTS

I would first and foremost like to acknowledge the math faculty members with which I have had the pleasure of attending classes, namely Dr. Friedman, Dr. Hale, Dr. Pulapaka, Dr. Coulter, and Dr. Szecsei. You all are responsible for creating my desire to make math a major, making the quest as painless as possible, and being there to share in the wonderful experiences. For that I am deeply obliged.

I would also like to especially thank Dr. Friedman for teaching me that I want to teach and for helping me more than expected or required in ways not expected or required.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

GENERALIZED CHESS ENDGAMES

By

Bruce Allan Moser

December 2003

Advisor: Dr. Erich Friedman
Department: Mathematics and Computer Science

We consider endgame scenarios of a variation of chess in which the object is to capture all of the opponent’s pieces. We find algorithms for capturing a single piece by the smallest number of a type of piece. To do so, we prove that a given number of a piece can capture the opponent's piece, and that fewer cannot.

CHAPTER 1
INTRODUCTION

The game of Chess has long been a favorite among people of all types, from the amateur searching for fun to the master searching for better strategies. The game itself is quite complex, but is built on the simple principles of approaching, threatening, and capturing pieces until the desired end is reached.

We will look at those simple principles more closely. We will consider endgame scenarios of a variation of chess in which the purpose of the game is to capture all the pieces of one's opponent, thus placing no special importance on a king and avoiding stalemates. Looking at only endgames in which white has 1 piece and black has 1 type of piece, we find algorithms for attacking a piece with another and the minimum number of pieces needed to assure victory.

1.1. METHOD

We determine the minimum number of a certain piece it will take to assure victory over one opposing piece, or find upper and lower bounds on this number. Since 5 different pieces can be used for offense or defense (pawns are excluded), there are actually 25 separate problems.

Our rules are as follows: The offensive pieces are first placed on the board in any desired configuration; the defensive piece is then placed. The offense has first move. This is a simplification of endgames that could occur in this generalization of chess, but still giving the overall flavor of endplay. To prove upper bounds, we find an algorithm for the offensive pieces that would assure the capture of the defensive piece. To prove lower bounds, we find an algorithm for the defensive piece that avoids capture. Throughout, we assume that the enemy will not move into a threatened space unless forced to.

1.2. TABLE OF RESULTS

The table below lists the values for the problems, with proofs in the later sections. The pieces along the top are the single defensive pieces to be captured by a number of the offensive pieces listed along the side. The entries are the minimum number of offensive pieces needed to assure victory. When only an upper bound is known, it is indicated with a ≤ sign; non-trivial lower bounds are known for some of the problems, and are indicated with a ≥ sign.

Defensive Piece

Offensive Pieces

	King	Queen	Knight	Rook	Bishop
King	≥ 2, ≤ 4	≥ 4, ≤ 8	≥ 2, ≤ 4	≥ 4, ≤ 7	≥ 3, ≤ 6
Queen	1	≤ 5	1	≤ 4	2
Knight	4	≤ 12	≤ 4	≤ 12	≤ 12
Rook	2	≥ 7, ≤ 8	≤ 8	7	≤ 4
Bishop	≥ 3, ≤ 4	≤ 8	≤ 4	≤ 8	≤ 8

1.3. DEFINITIONS

Each piece has a unique style in which it can move. A king is permitted to move into any neighboring horizontal, vertical or diagonal square (figure 1.3a); a queen can move into any square along the queen's column, row, or either of the diagonals (figure 1.3b). A rook moves into any square along its column or row (figure 1.3c), a bishop into any square along its diagonals (figure 1.3d), and a knight into any square which is either one row and two columns or one column and two rows away (figure 1.3e). When a piece moves the same length and in exactly the same direction as another, we call it *imitation*.

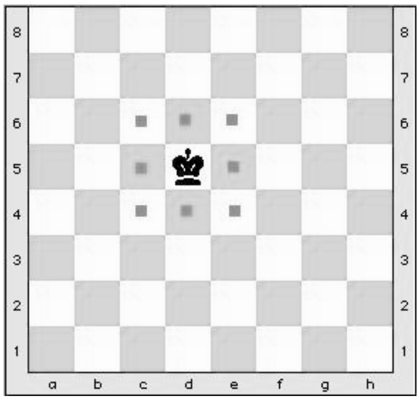

figure 1.3a: king's moves

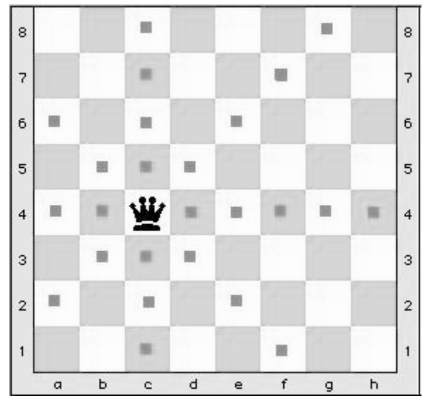

figure 1.3b: queen's moves

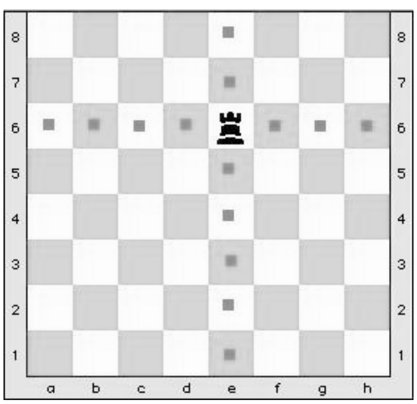

figure 1.3c: rook's moves

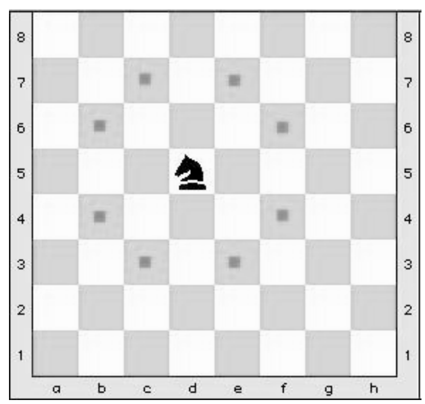

figure 1.3d: knight's moves

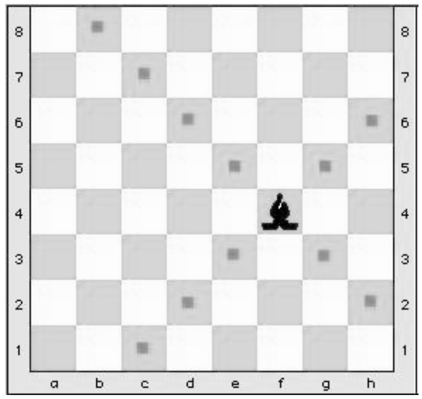

figure 1.3e: bishop’s moves

The chess board is made up of 64 squares in an 8×8 configuration which can be classified into 3 types (see figure 1.3f): A *middle* square is any square bordered by exactly 8 others; an *edge* is a square bordered by exactly 5 others; a *corner* is bordered by exactly 3 others.

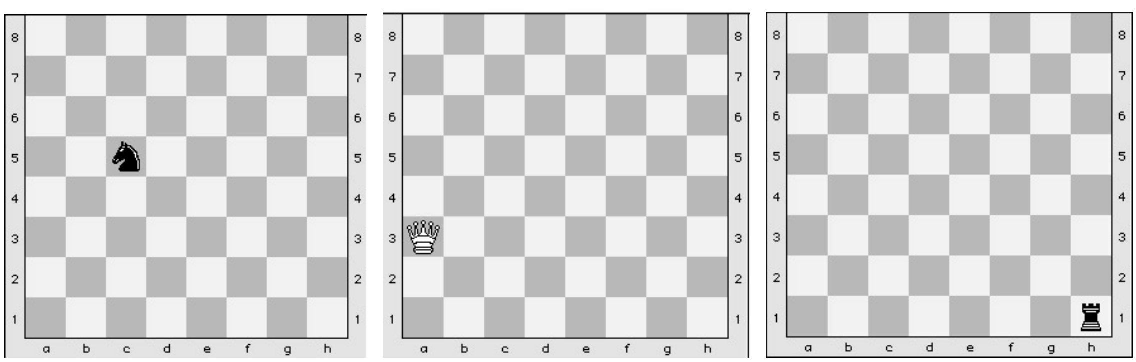

figure 1.3f: examples of a middle square, edge and corner, respectively

CHAPTER 2
CAPTURING WITH KINGS

2.1. TO CAPTURE A KING WITH KINGS

Although we believe a king can be captured by using as few as 2 kings, the algorithm used with 4 kings proves to be important in more than one instance.

Theorem 2.1.1. 4 kings can capture a king

Proof. The offensive kings are first placed in the configuration shown in figure 2.1.1a.

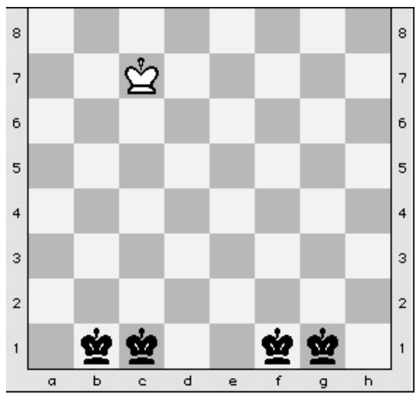

figure 2.1.1a: starting positions of kings against king

Then they are moved by repeating the following algorithm.

- Move the king in column B up one square.
- Move the king in column C up one square.
- Move the king in column F up one square.
- Move the king in column G up one square.

In each of the four possible configurations of the attacking kings (see figure 2.1.1b), the threatened squares contain dots, making it clear to see that they prohibit any movement by the opposing king below the highest king. We call this technique the "Roman army" approach. Each move by the attacking kings diminishes the number of safe squares left above the kings for the opposing king to move around in. This number is strictly decreasing, so it will eventually decrease to 0 when the kings all reach row 7, at which time the opposing king is forced to move into a guarded square. □

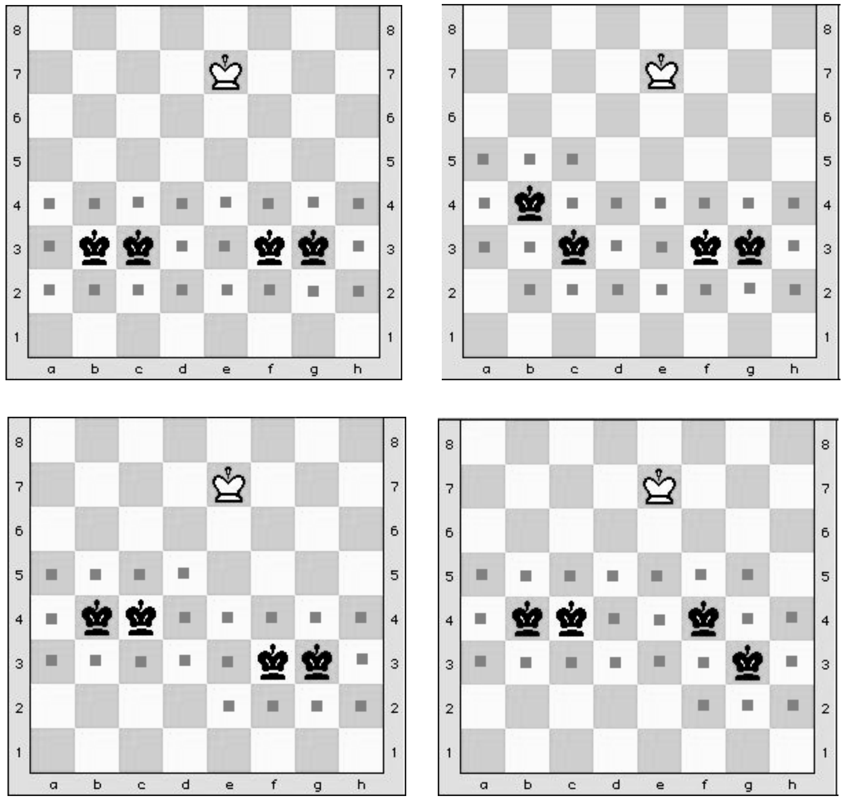

figure 2.1.1b: 4 configurations of attacking kings and controlled squares

Theorem 2.1.2. 1 king cannot capture a king

Proof. Due to the threatening ability of a king, 1 king cannot capture another. When the defensive king is in a middle square, he has 8 possible squares into which to move; a defensive king can safely threaten at most 3 of these. The defensive king on an edge has 5 possible moves, a maximum of 3 which can be safely threatened by the offensive king. And when the defensive king is in a corner, he has 3 moves, only 2 of which can be safely threatened. □

2.2. TO CAPTURE A QUEEN WITH KINGS

Theorem 2.2.1. 8 kings can capture a queen.

Proof. The offensive kings are placed along the left edge, as shown in figure 2.2.1a.

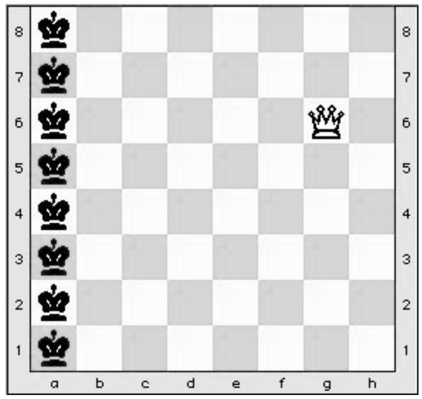

figure 2.2.1a: starting positions for kings against queen

Let king #1 be the king in row 1, king #2 in row, and so on, and let the kings be in column *k*. Let their movement be limited to horizontal. The following algorithm is then used.

- If the queen is threatening to diagonally move through the kings, move the king in the row of the hole one square to the right into column *k+1*.
- Otherwise, if the lowest-numbered king in column *k* is not attacked on the diagonal, move it right one square into *k+1*.
- Otherwise, move the second lowest-numbered king right one square into column *k+1*.

This "Roman army" approach prevents the queen from zooming through the kings onto the other side of the board for the following reasons. Firstly, since a king threatened on a diagonal is never moved, a hole is never opened up through which the queen could immediately move (figure 2.2.1b). Secondly, since a queen cannot simultaneously threaten 2 parallel diagonals, a maximum of 1 king can be "skipped" – left in column *k* when the king above has moved into column *k+1.*

After this move, another king cannot be skipped because of the order of the algorithm: for another king to be skipped, the queen would have to not be threatening to move through a hole and threatening the lowest-numbered king in column *k* at the diagonal; it is impossible to threaten the lowest-numbered king in column *k* at the diagonal when a king has been skipped (the lowest-numbered king is the skipped king), and thus the skipped king will move into column *k+1* before any other kings can be skipped. Therefore, since the queen would safely be able to threaten at most one diagonal opening in the army of kings, the threat of zooming through an opening in the kings can always be blocked (figure 2.2.1c). (The only place where the queen could attack two diagonals is threatened by the kings.)

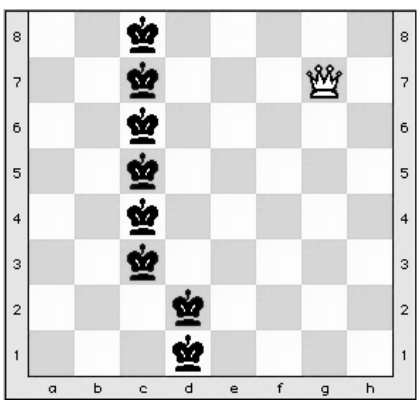

figure 2.2.1b: king #3 to be skipped

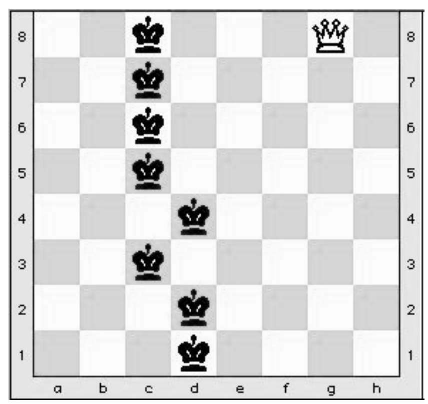

figure 2.2.1c:, king #5 to move

Since the queen is not permitted to pass through the pursuing kings, each move by the kings decreases the number of free squares into which the queen can move. Thus, the number of free square for the queen is strictly decreasing and will go to 0 in finite time. □

Although we believe that 4 kings cannot capture a queen, a suitable algorithm was not found. We therefore prove that 3 kings cannot capture a queen.

Theorem 2.2.2. 3 kings cannot capture a queen.

Proof. Since the queen can move horizontally and vertically similar to a rook, the algorithm in theorem 2.4.2 is sufficient to prove that 3 kings cannot capture a queen. □

2.3. TO CAPTURE A KNIGHT WITH KINGS

Theorem 2.3.1. 4 kings can capture a knight.

Proof. As stated before, the method of attack used by 4 kings proves to be a powerful one. By using the "Roman army" algorithm found in section 2.1.1, the 4 kings can assure victory over a knight as well, due to the fact that the attacking method at all times guards every square on two consecutive rows. Only able to clear two rows, a knight's move is not sufficient to pass over the attacked squares. □

Theorem 2.3.2. 1 king cannot capture a knight.

Proof. Due to the threatening ability of a king, 1 king is not sufficient to capture a knight. A knight in a middle square has at least 6 moves; only 2 of these moves can be threatened by a king. When on an edge, a knight has at least 3 moves, only 2 of which can be threatened by a king. And although a knight in a corner has only 2 moves, a king cannot force him to move there. In all cases but 1, the knight has more than 1 non-threatened square into which to move. In the case when the knight is in a square on an edge and next to a corner, he has only 3 moves, 2 of which can be threatened; however, after moving to the non-threatened square, he will again have at least 4 possible moves. □

2.4. TO CAPTURE A ROOK WITH KINGS

Although we believe 6 kings to be sufficient to capture a rook, an algorithm is difficult to find. Thus, we prove that 7 are enough.

Theorem 2.4.1. 7 kings can capture a rook.

Proof. Similar to the algorithm used in theorem 2.1.1, the kings will employ the "Roman army" approach, this time on the diagonal. To begin, arrange the 7 offensive kings along the middle diagonal, leaving the bottom right corner empty (see figure 2.4.1a). The following algorithm can then be used. Let the kings be in diagonal *k*, where diagonal *k* is the top-left to bottom-right diagonal that contains the k^{th} square of row 8. Let the kings be numbered according to their row number, and let the movement of the kings be limited to horizontal.

- Move the highest-numbered king in diagonal *k* not on the right edge right one square into diagonal *k+1*.

By moving 1 king at a time horizontally to the right, no horizontal or vertical holes are made in the Roman army of kings. And although H1 is open, it is threatened and impossible to move into after king #2 moves to the right. □

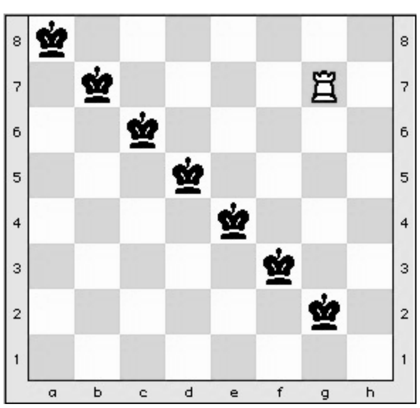

figure 2.4.1a: starting configuration

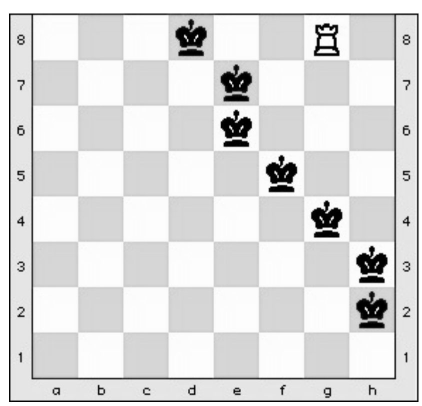

figure 2.4.1b: sample configuration

Theorem 2.4.2. 3 kings cannot capture a rook.

Proof. To prevent 3 kings from capturing a rook, we restrict the movement of the rook to the 9 squares shown in figure 2.4.2a. The rook is first placed in any one of these 9 squares not threatened by a king; a non-threatened square exists because each king can threaten at most 1 of the possible starting squares. If the kings are not in the rook's direct path, they can threaten at most 3 of the rook's possible moves. Due to the configuration of the possible moves, the rook always has 4, and thus, has a move not threatened by the kings. If a king is in the rook's path, he can deny 2 of the rook's possible moves, but in order to be in the rook's path, another king must be in an adjacent square to protect it and is not able to threaten another of the rook's possible moves. Thus, the rook has at least one free move. If 2 kings are in the rook's path, and both are protected by the 3rd king, they can capture the rook (see figure 2.4.2b). However, this configuration is impossible; for it to occur, the rook must move into the threatened corner square when a non-threatened move exists. □

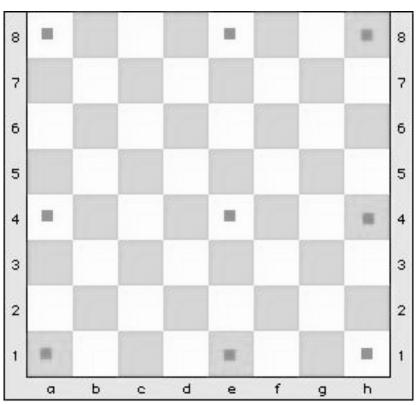

figure 2.4.2a: possible moves by rook

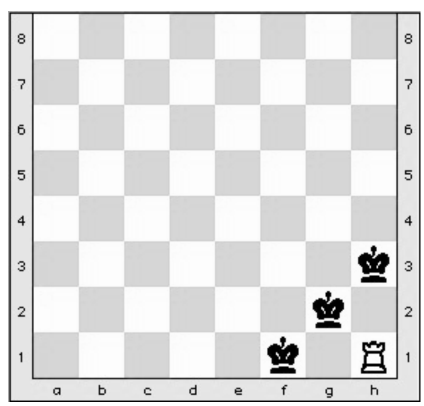

figure 2.4.2b: impossible situation

2.5. TO CAPTURE A BISHOP WITH KINGS

Theorem 2.5.1. 6 kings can capture a bishop.

Proof. The algorithm used is another application of the "Roman army" approach. The kings are similarly placed along an edge, leaving the corner squares free (see figure 2.5.1). Again, let the king in column 2 be king #2, etc., let the kings be in column *k*, and let the movement be horizontal.

> - If the bishop is threatening to diagonally move through the kings, move the king in the row of the hole one square to the right into column *k+1*.
> - Otherwise, if the lowest-numbered king in column *k* is not attacked on the diagonal, move it right one square into *k+1*.
> - Otherwise, move the second lowest-numbered king right one square into column *k+1*.

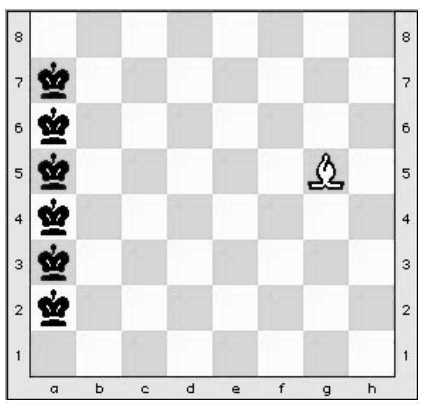

figure 2.5.1: starting position

The justification that this algorithm works is similar to that found in Theorem 2.2.1. with one addition: since the bishop cannot move horizontally, kings in rows 1 and 8 are not necessary; kings #2 and #7 threaten the squares in rows 1 and 8 to prevent the bishop from passing through the army of kings. □

Theorem 2.5.2. 2 kings cannot capture a bishop.

Proof. In order for a bishop to escape 2 kings, the bishop is first placed into 1 of the 4 squares shown in figure 2.5.2. The following algorithm is then used.

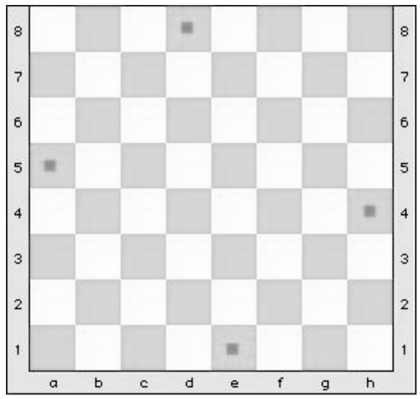

figure 2.5.2: possible starting squares for the bishop

- If possible, move the bishop along the diagonal with 4 squares to the edge.
- If this move is threatened or denied, move along the diagonal with 5 squares to the edge.
- If this move is threatened, move 1 square along the diagonal with 5 squares, and move back next turn.

For the move along the diagonal with 4 squares to be impossible, it must be either threatened or denied. If a king moves into the diagonal with 4 squares and thus denies the move, the other king must be in an adjacent square thus leaving the square at the end of the diagonal with 5 squares unthreatened. If the king does not move into the diagonal with 4 squares but simply threatens the square at the end, the other king could be threatening the square at the end of the diagonal with 5 squares and thus both moves would be threatened. In this case, the 3rd step of the algorithm is used; the bishop temporizes by moving one square inward into a square not threatened by either king. (2 kings cannot threaten both squares at the end of the diagonals and the square into which the bishop moves to temporize). The bishop then safely moves back and the escape algorithm is repeated. □

CHAPTER 3
CAPTURING WITH A QUEEN

When dealing with queens, a trivial upper bound for all the pieces can be found by placing queens on the board in such a way as to attack every square, which we call the "Attack Everything in Sight" method. As proved by Steinhaus in 1999 [1], the minimum number of queens needed to attack every other square is 5 (see figure 3).

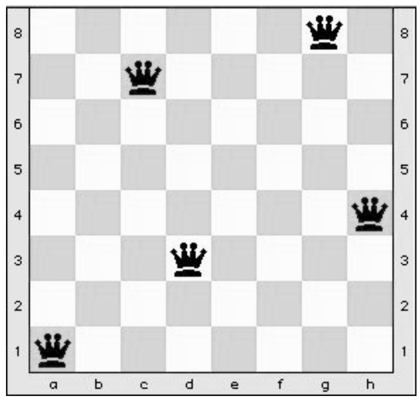

figure 3: 5 queens attack every square

3.1. TO CAPTURE A KING WITH QUEENS

Theorem 3.1. 1 queen can capture a king.

Proof. The queen is first placed in B1. In order to capture a king, the queen must first move into a square a knight's move away from the king. To accomplish this, the following algorithm is used.

- If the king starts in C2, move the queen two spaces up.
- If the king starts in row 2 but not in C2, move the queen horizontally until she is a knight's move from the king.
- If the king does not start in row 2, move the queen into his column. After he moves, move the queen up until she is a knight's move away.

The algorithm moves the queen a knight's move away in each case:

- If the king starts in D2, moving 2 spaces up clearly moves the queen a knight's move away from the king.
- If the king starts in another square on row 2, there is at least 1 square a knight's move away from the king in row 1 into which the queen could move.
- If the king starts somewhere else, moving the queen into his column forces the king move from that column. After this move, there is at least 1 square a knight's move away from the king in her column into which the queen could move.

Now that the queen is a knight's move away from the king, the following algorithm can be used for his capture.

> • If the king moves in such a way that imitation is possible, imitate his move with the queen.
> • If the king moves in such a way that his imitation is not possible, move the queen two spaces toward the middle.

To prove that this algorithm traps the king in finite time, we show that each move by the queen (with the exception of one) makes progress towards his capture. To do so, we number the chess board and, without loss of generality, let a certain configuration be the canonical one, as shown in figure 3.1a (when the queen is a knight's move away from the king, this configuration can be obtained by rotating and/or reflecting the board as necessary).

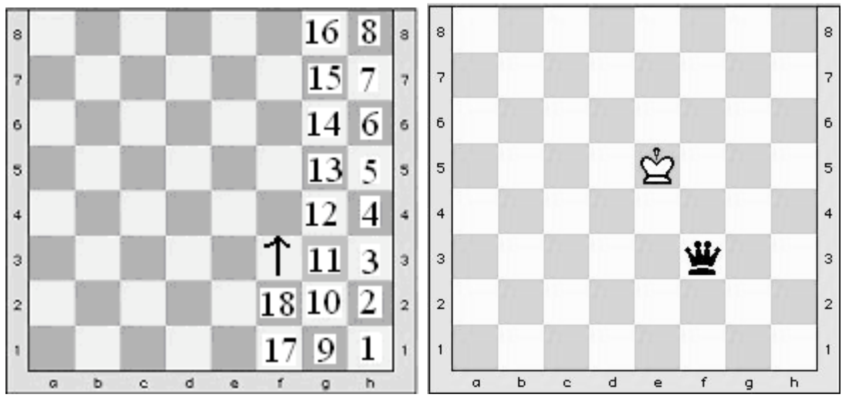

figure 3.1a: numbered chess board and canonical configuration

We now consider the possible moves by the king: up, up and left, or down and left. In each case, the queen can imitate. By numbering the board as shown in the figure, each of the queen's moves is into a larger-numbered square (in figure 3.1a, the queen, in square #19, could imitate the king's move into square #20, 26, or 28).

A certain circumstance, however, prohibits the queen from imitating the king's move. This occurs when the queen reaches the bottom edge and the king still can move down and left. Shown in figure3.1b are the squares in which the queen must be originally for this special case to occur.

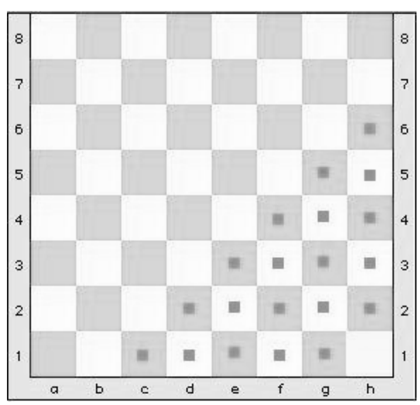

figure 3.1b: possible special case squares

When the queen is in one of these squares and the king continues to move in the down and left direction, the queen will be forced to use the second step of the algorithm and move up two squares (figure 3.1c).

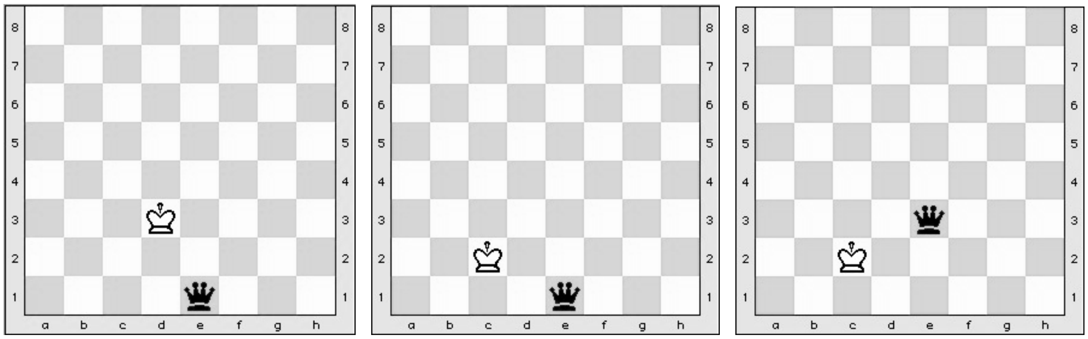

figure 3.1c: special move by the queen

This move, however, can occur at most once; figure 3.1d shows all possible scenarios of the special move. The queen starts from 1 of the 5 squares indicated in the left figure, ends in 1 of the squares in the middle figure, and after rotation ends in 1 of the squares in the right figure. This shows that after the queen has moved two squares up and the board has been rotated and renumbered, the queen is no longer in one of the squares shown in figure 3.1b which could result in the special scenario.

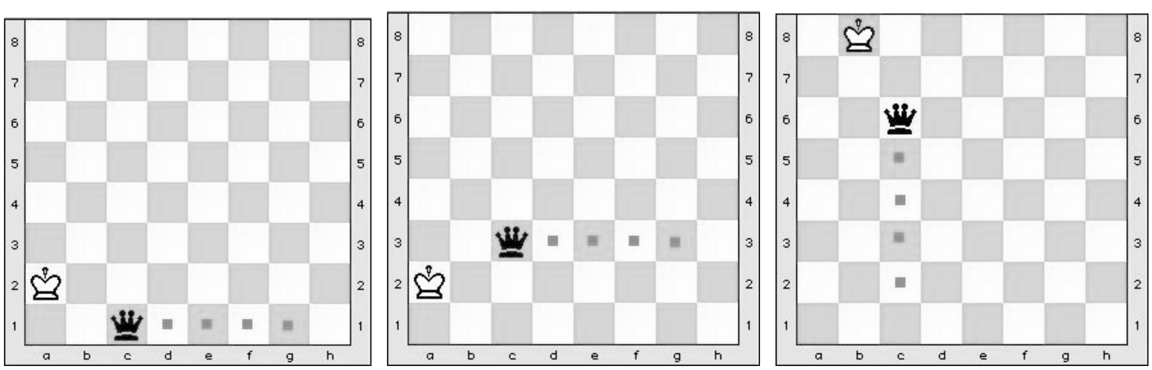

figure 3.1d: all special move scenarios and rotation

With the exception of the special case, which can happen at most once, each move by the queen makes progress, i.e. the queen moves into a square with a larger number. Thus, the queen will eventually push the king into the corner, forcing him to move into a threatened square. □

3.2. TO CAPTURE A QUEEN WITH QUEENS

Using the "Attack Everything in Sight" method, we know that 5 queens are sufficient.

3.3. TO CAPTURE A KNIGHT WITH QUEENS

Theorem 3.3. 1 queen can capture a knight.

Proof. To show that a queen can capture a knight, we place the queen in square D4 (figure 3.3a) and consider 3 cases.

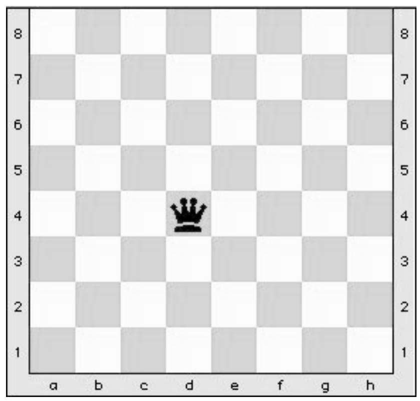

figure 3.3a: starting configuration

Case 1 – Knight starts on an edge.

Without loss of generality, the knight must be in row 1 or column H. If he is in row 1 (including square H1), the queen can move horizontally until she is directly above the knight, 3 spaces away; this threatens all of the knight's next moves. If the knight is in column H, we have 1 of 4 cases:

- if the knight begins in H3 or H5, the queen can move 3 squares to the left of the king, threatening all of the knight's next moves;
- if the knight begins in square H7, the queen moves diagonally into square G7 thus threatening all of the knight's next moves;
- if the knight is in H2, the queen moves two squares right into F4 and threatens

the next moves;

- if the knight is in H6, the queen again moves two square right into F4 and after the knight's move into G8, the queen moves into G5, threatening his next moves.

<u>Case 2</u> – Knight starts one row or column inside the edge.

Without loss of generality, the knight must be in row 2 or column G. If he is in row two (squares C2, E2 or G2), the queen can move into C3, E3 or F2 respectively and threaten all of the knight's next moves. If the knight is in column G, we have 1 or 3 cases:

- if the knight is in G6, the queen moves into F6 and threatens all of the next moves;
- if the knight is in G3, the queen moves to G4; if the knight then moves into H1, the queen moves to H4 and threatens the next moves; if the knight decides to rather move into F1, the queen moves to F4 and threatens the next moves;
- if the knight is in G5, first, the queen moves into G4; if the knight moves into G7, the queen moves into F7 and threatens the next moves; if the knight moves, rather, into E7, the queen moves into E4; if the knight then moves into H8, the queen moves into E6 and threatens all next moves; if the knight moves, rather, into D8, the queen moves into E4 and forces the knight to B7; the queen then happily moves to B6 and threatens all of his next moves (see figure 3.3b).

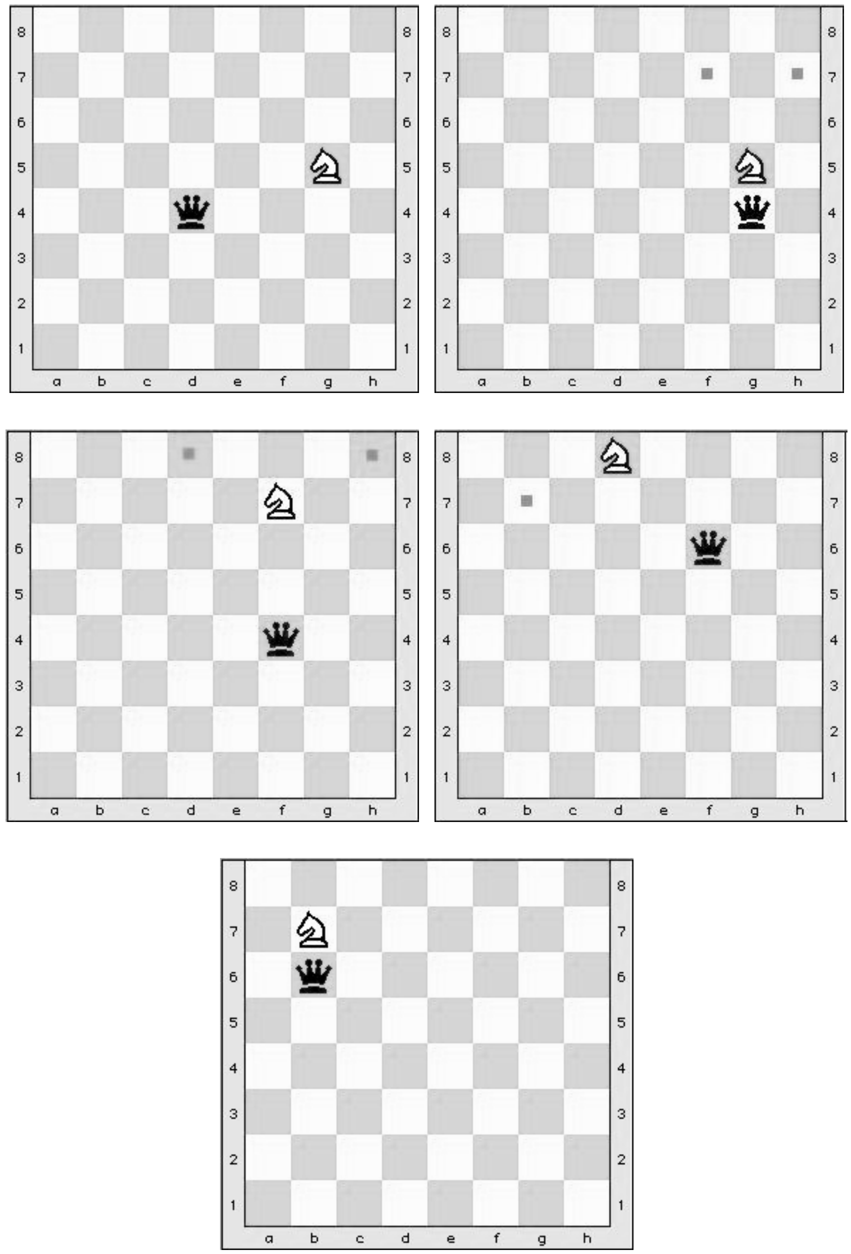

figure 3.3b: case 2, knight starting in G5

<u>Case 3</u> – the knight starts in C6, E6, F5 or F3.

Since the squares C6 and E6 are the reflections of F5 and F3, we handle just those 2 squares. If the knight starts in F5 or F3, the queen moves into E5 or E3, respectively, forcing the knight to the edge. The queen then moves vertically into the square three spaces left of the knight, threatening all of the knight's next moves. □

3.4. TO CAPTURE A ROOK WITH QUEENS

Theorem 3.4.1. 4 queens can capture a rook.

Proof. The queens are first placed in the center of the board, as shown in figure 3.4.1a. There are 8 non-threatened squares in which the rook could be; without loss of generality, let the rook be in square F1. The following algorithm is then used.

- Move the queen in E5 into F4.

This move by the queens threatens all of the possible moves of the rook, thus assuring its capture (figure 3.4.1b). □

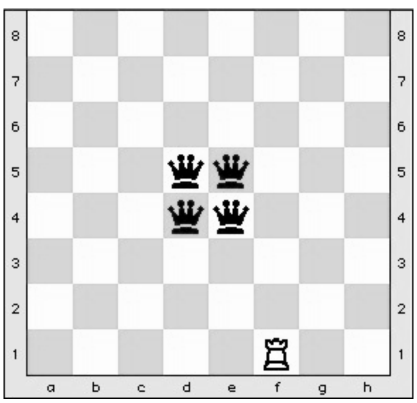

figure 3.4.1a: starting configuration

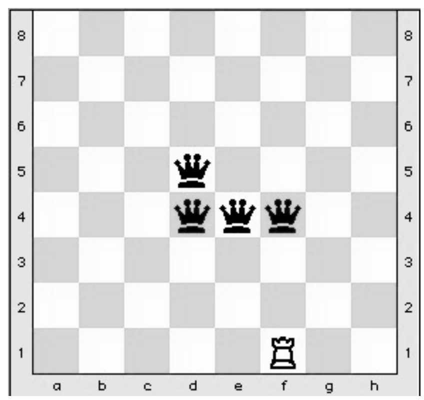

figure 3.4.1b: move by queens

3.5. TO CAPTURE A BISHOP WITH QUEENS

Theorem 3.5.1. 2 queens can capture a bishop.

Proof. The queens are first placed in the center of the board (figure 3.5.1a).

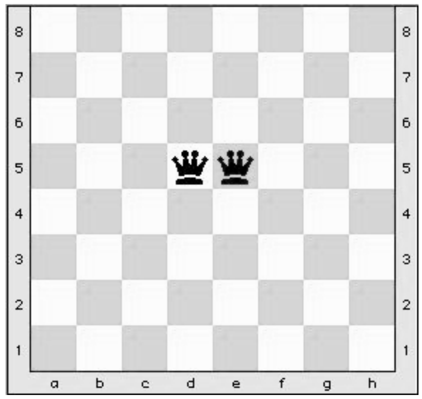

figure 3.5.1a: starting configuration

The following 2 algorithms are then used.

- If the bishop begins on a white square, move the queen in E5 to E4. Otherwise, move the queen in D5 to D4.

This algorithm simply moves the queens on the same color as the bishop, thus increasing their power. After this is accomplished, the bishop must move into 1 of 4 safe squares (figure 3.5.1b shows these squares if the bishop started on white). Without loss of generality, let the bishop be in square F1. The following algorithm is then used.

- Move the queen in E4 to E6.

After this move, all the possible moves by the bishop are threatened by the queens (figure 3.5.1c). □

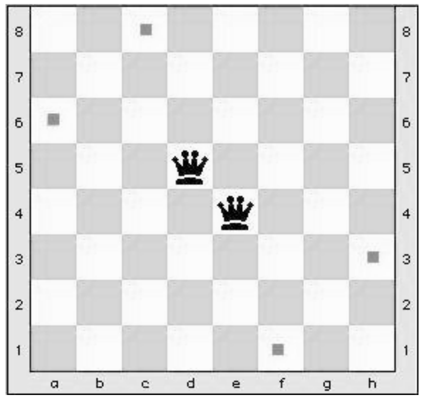

figure 3.5.1b: 4 safe squares

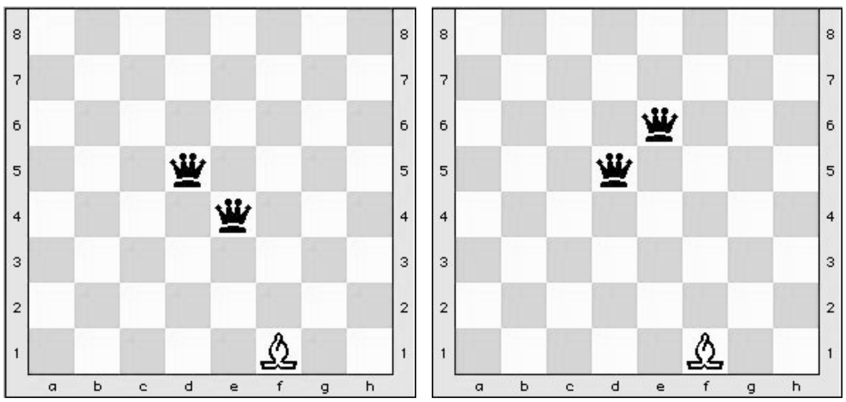

figure 3.5.1c: sample move by queens

Theorem 3.5.2. 1 queen cannot capture a bishop.

Proof. Due to the threatening ability of a queen, 1 is not sufficient to capture a bishop. If the bishop is in a middle square, he has at 9 squares into which he can move; these squares cannot all be guarded by the queen. A bishop on an edge has 7 moves, all of which cannot be threatened by a queen. In the corner, a bishop has 7 moves; in order to threaten these 7 squares, the queen must be on that diagonal, thus in danger of capture. □

CHAPTER 4
CAPTURING WITH A KNIGHT

Just as 5 queens are sufficient to attack each square, 12 knights, proven by Dudeney [2], also serve as an upper-bound (figure *x*).

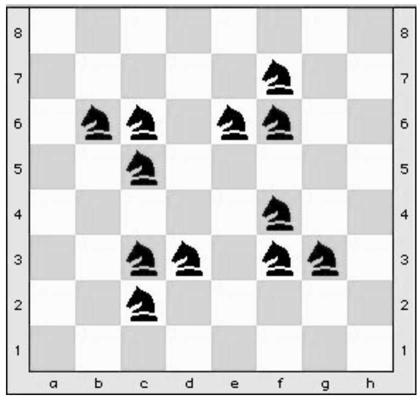

figure 4: 12 knights attacking every square

4.1. TO CAPTURE A KING WITH KNIGHTS

Theorem 4.1.1. 4 knights can capture a king.

Proof. The offensive knights are placed in the configuration shown in figure 4.1a. Without loss of generality, this configuration leaves 5 free squares for the king, as shown by the marked squares.

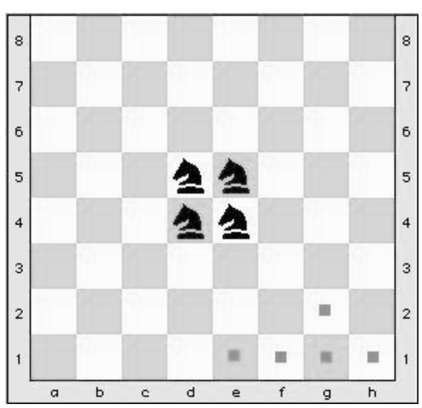

figure 4.1.1a: starting configuration and non-threatened squares

The following algorithm is then used, shown in figure 4.1b. (The reflection or rotation of the algorithm should be used if the king is in the other squares).

- Move the knight in D5 to E3.
- Then, if the king is in E1, move the knight in E5 to G6. Otherwise, move the knight in D4 to F3
- Then, if the king is in H1, move the knight in E5 to G6.

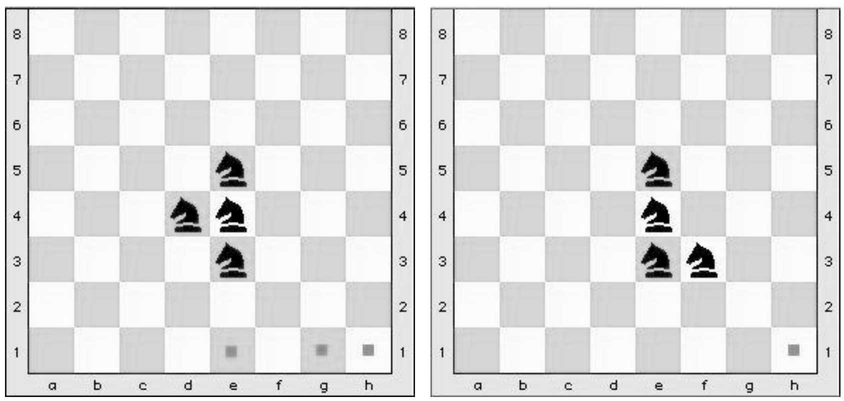

figure 4.1.1b: steps one and two in the algorithm

After moving the knight from D5 to E3, there are 3 non-threatened squares for the king. If the king is in E1, all 5 possible next moves are threatened, and thus the algorithm instructs a move of the knight in E5 to G6 – in other words, it moves the non-crucial knight so that the king is forced to move into a threatened square. If the king is in one of the other two non-threatened squares (G1 or H1), the knight in D4 is moved to F3 in order to threaten G1. If the king were previously in H1, he would then be forced to move into a threatened square; if he were in G1, he would now be forced to move into H1. Thereafter, the knight in E5 would then be moved simply to force the king to move into one of the three threatened squares around him. □

Theorem 4.1.2. 3 knights cannot capture a king.

Proof. Since a knight can only threaten at most 2 of a king's 8 possible moves, 2 knights can threaten at most 4. Thus, a king always has at least 4 possible unthreatened moves. Because of this, the king can remain in a middle square, never forced into a corner. □

4.2. TO CAPTURE A QUEEN WITH KNIGHTS

Using the "Attack Everything in Sight" method, we know that 12 knights are sufficient.

4.3. TO CAPTURE A KNIGHT WITH KNIGHTS

Theorem 4.3.1. 4 offensive knights can capture a knight.

Proof. The knights are first placed in the middle as in section 4.1. Figure 4.3a shows this configuration and the non-threatened squares (without loss of generality) where the defensive knight could be.

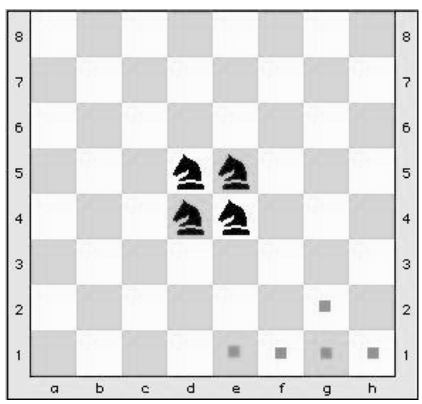

figure 4.3.1: the 4 knights and the non-threatened squares

The algorithm for capture is as follows (The reflection or rotation must again be used if necessary):

- If the defensive knight starts in E1, move the knight in D5 to E3.
- If the defensive knight starts in F1, move the knight in E5 to G4.
- If the defensive knight starts in G1, move the knight in D5 to F4.
- If the defensive knight starts in H1, move the knight in E5 to G6.
- If the defensive knight starts in G2, move the knight in D4 to C2, then move the knight in D5 to E7, then move the knight in E7 to G6.

If the knight is in one of the 4 non-threatened edge squares, all of the possible squares corresponding to his next move can be threatened simply by moving the appropriate piece following the algorithm. When, however, the defensive knight is in G2, all of his moves cannot be covered. By first moving the offensive knight in D4 to C2, the move into E1 is threatened and the defensive knight is forced to move into H4. Moving the offensive knight in D5 to E7 threatens G6, forcing the defensive knight to move back into G2. By then moving the offensive knight in E7 to G6, all of the moves by the defensive knight are covered and he must move into a threatened square. □

4.4. TO CAPTURE A ROOK WITH KNIGHTS

Using the "Attack Everything in Sight" method, we know that 12 knights are sufficient.

4.5. TO CAPTURE A BISHOP WITH KNIGHTS

Using the "Attack Everything in Sight" method, we know that 12 knights are sufficient.

CHAPTER 5
CAPTURING WITH A ROOK

5.1. TO CAPTURE A KING WITH ROOKS

Theorem 5.1.1. 2 rooks can capture a king.

Proof. The rooks are placed in the corner as shown in figure 5.1.1a. Let the rook in square H1 be named 'rook #1' and the rook in G1 'rook #2.' The following algorithm, relatively unknown to chess players [4], is then used.

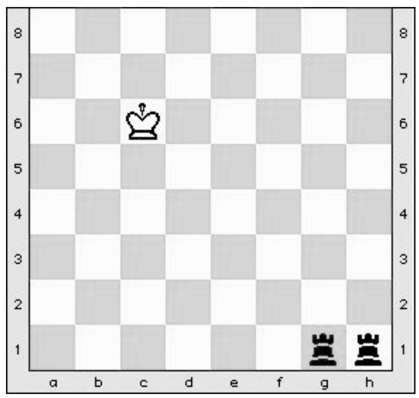

figure 5.1.1a: starting position for rooks against king

- If the king is not in the row above rook #1, move rook #1 up 1 square (step 1 in figure 5.1.1b).
- Otherwise, if the king is not in the column to the left of rook #2, move rook #2 into that column.
- Otherwise, if the king is in both the row above rook #1 and the column to the left of rook #2, move rook #2 either up one square if it is in row 1 or down one square if it is in row 2.

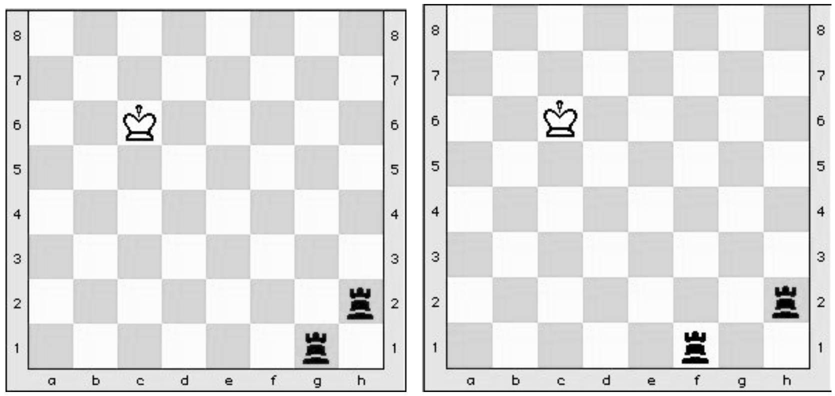

figure 5.1.1b: two moves by the rooks

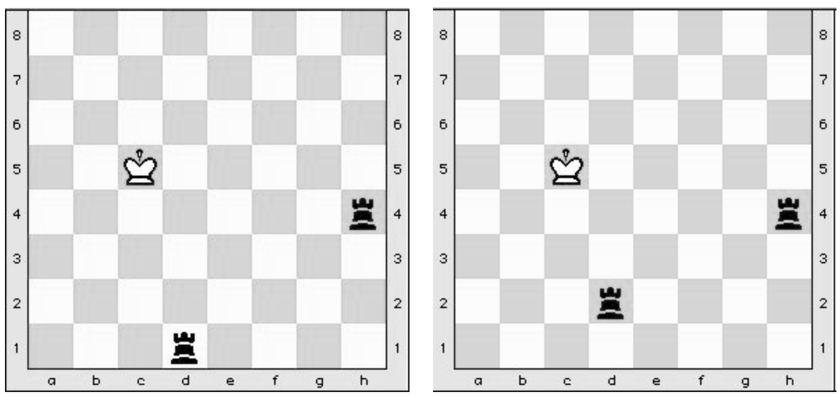

figure 5.1.1c: third step of algorithm

A certain starting position poses problems, namely, when the king begins in square F1 (figure 5.1.1d).

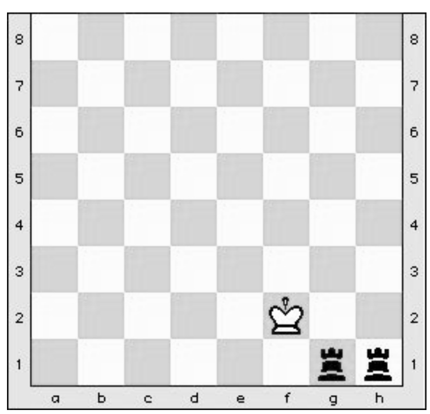

figure 5.1.1d: special case starting position

In this case, neither rook can move according to the algorithm, thus, the following steps must first be used (figure 5.1.1e).

- Move rook #2 across the board into square A1.
- After the king makes his move, move rook #1 across the board into square B1.

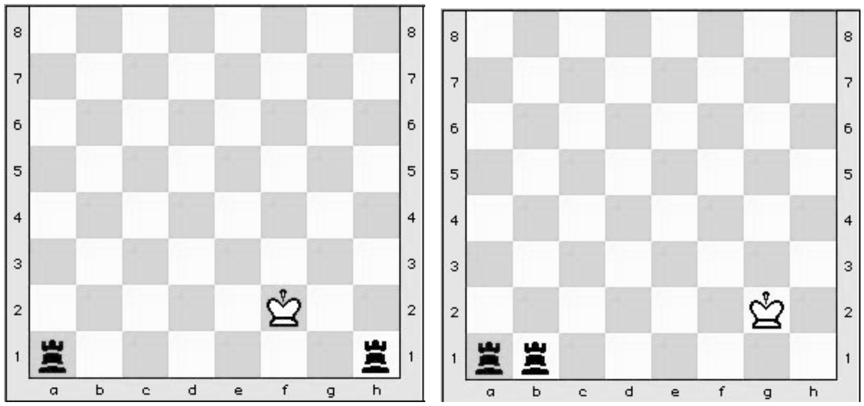

figure 5.1.1e: opening moves for the special case starting position

The rooks are now in position to use the mirror-image of the algorithm.

Each move, with one exception of the 3rd step in the algorithm, decreases the number of free squares by cutting off a row or column. The 3rd step does not decrease this number; however, it does not increase the number and the move following it does decrease the number, since the king will be forced to move either up, left, or up-and-left, and therefore open up a row or column for the algorithm. Also, since a threatened row and column separates the king from the rooks at all time, the rooks will avoid capture. Thus, by following the previous algorithms, the number of free squares into which the king can move will decrease to 0, at which time the king will be forced to move from the corner into a square guarded by a rook. □

Theorem 5.1.2. 1 rook cannot capture a king.

Proof. Due to the guarding ability of a rook, it is easy to show that a king cannot be captured. If the king is in a middle square, he has 2 possible rows and 2 possible columns into which to move; since a rook can only guard one column and one row, the king cannot be captured in a middle square. When the king is on an edge, he has 1 possible row and 2 possible columns into which to move; since a rook can only guard one column and one row, the king cannot be captured on an edge. And if the king is in a corner, he has 1 possible column and 1 possible row into which to move, but in order for the rook to cover both the column and row, he would have to be within one space of the king, thus putting the rook within capturing distance of the king. □

5.2. TO CAPTURE A QUEEN WITH ROOKS

Using the "Attack Everything in Sight" method, 8 rooks are sufficient.

Theorem 5.2.2. 6 rooks cannot capture a queen.

Proof. As shown in theorem 5.4.2, 6 rooks cannot capture another rook. Since a queen can move both horizontally and vertically like a rook, 6 rooks cannot capture a queen. □

5.3. TO CAPTURE A KNIGHT WITH ROOKS

Although we believe that 2 rooks can capture a knight, it is difficult to prove.

5.4. TO CAPTURE A ROOK WITH ROOKS

Theorem 5.4.1. 7 rooks can capture a rook.

Proof. Using the starting configuration shown in figure 5.4.1a, the following algorithm can be used.

- Move a rook in row 1 into a row not guarded by the enemy rook
- Repeat until one rook is left in row 1.

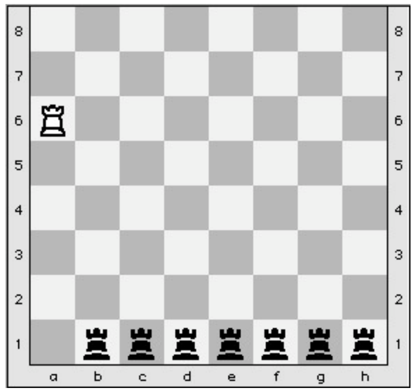

figure 5.4.1a: starting configuration for rooks against rook

The algorithm ends with the opposing rook forced to move into one of the 7 rows or columns guarded by the 7 rooks (figure 5.4.1b). □

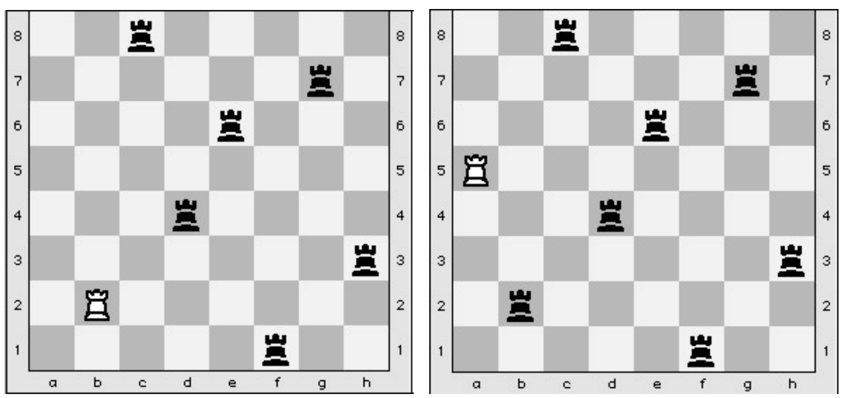

figure 5.4.1b: sample configuration

Theorem 5.4.2. 6 rooks cannot capture a rook.

Proof. Regardless of what configuration the defensive rooks are placed, the offensive rook will always have at least one empty row or column into which to move (figure 5.4.2).

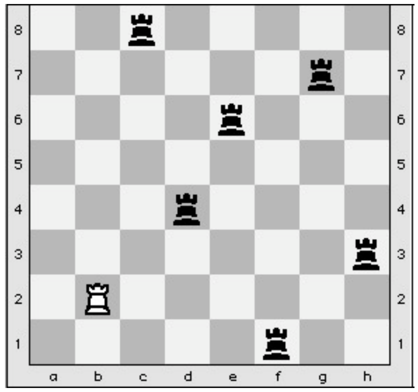

figure 5.4.2: 6 rooks leaving an empty row

5.5. TO CAPTURE A BISHOP WITH ROOKS

Theorem 5.5.1. 4 rooks can capture a bishop.

Proof. The rooks are first placed in the bottom row as shown in figure 5.5.1a.

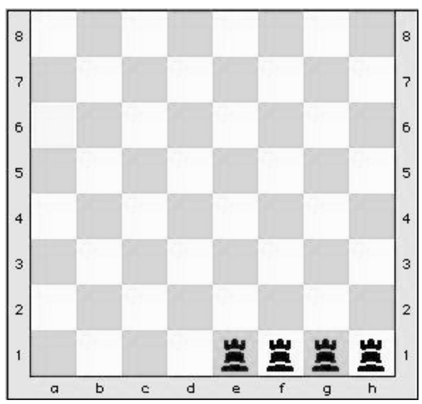

figure 5.5.1a: starting configuration

The following algorithm is then used.

- If the bishop starts on a black square, move the rook in column E up 1 square, the rook in column F up 2 squares, and the rook in column G up 3 squares. Otherwise, move the rook in column F up 1 square, the rook in column G up 2 squares, and the rook in column H up 3 squares.
- Move the rook in the lowest row up 4 squares. Repeat this step only.

After the first step in the algorithm (figure 5.5.1b), the 4 rooks are on squares not of the same color as the bishop and which threaten the bottom 4 rows. By continually moving the rook in the lowest row up 4 squares, the rooks remain on safe squares (since the bishop stays on one color) and move the four subsequent threatened rows upward. Since the bishop has only 4 columns to move around in, he can move *at most* 4 rows at a time; this prohibits him from moving through the threatened rows created by the Roman army of rooks. Therefore, each move by the rooks decreases the number of rows into which the bishop could move by 1, going to 0 after 7 moves. □

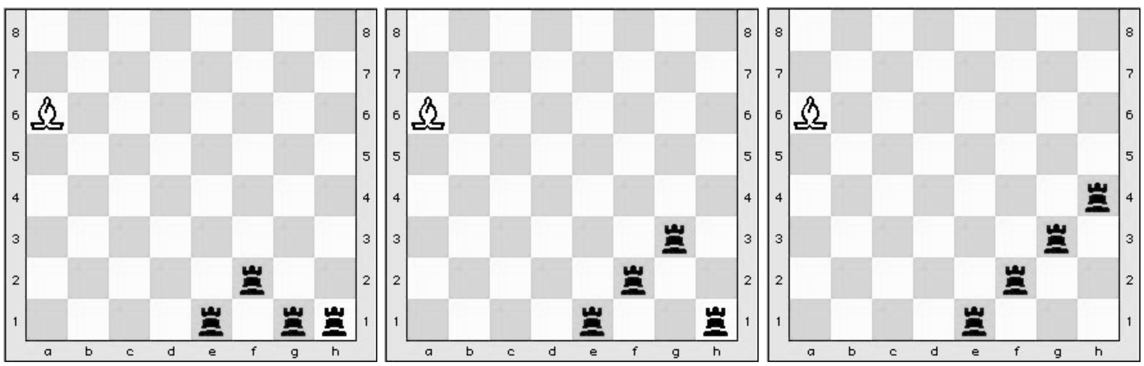

figure 5.5.1b: first step of algorithm if bishop starts on white

CHAPTER 6
CAPTURING WITH A BISHOP

8 bishops, as shown by Dudeney [3], is the minimum number of bishops needed to attack every square (figure 6), and thus 8 is an upper-bound on bishops for capturing any piece.

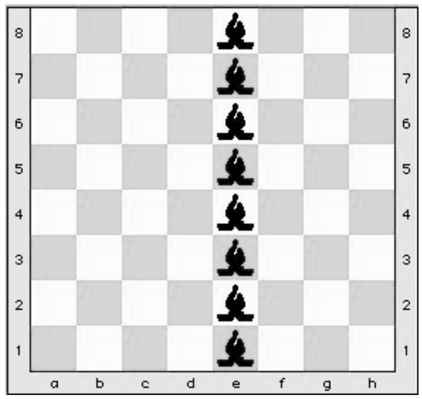

figure 6: 8 bishops attacking every square

6.1. TO CAPTURE A KING WITH BISHOPS

Although we believe a king can be captured with 3 bishops, the algorithm proves to be a difficult one. Using 4 bishops, however, makes the capture simple, serving as a useful algorithm for the capture of a knight as well.

Theorem 6.1.1. 4 bishops can capture a king.

Proof. The offensive bishops are first placed in the center of the board as shown in figure 6.1.1a.

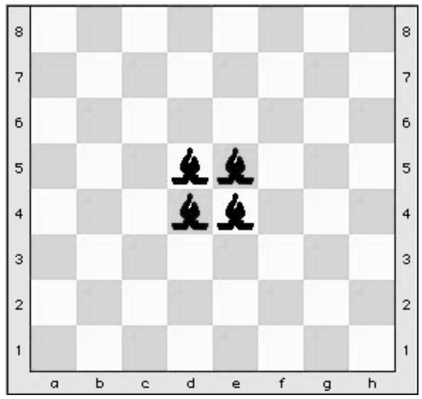

figure 6.1.1a: bishop starting configuration

Without loss of generality, the king could be in one of 6 non-threatened squares, as shown in figure 6.1.1b.

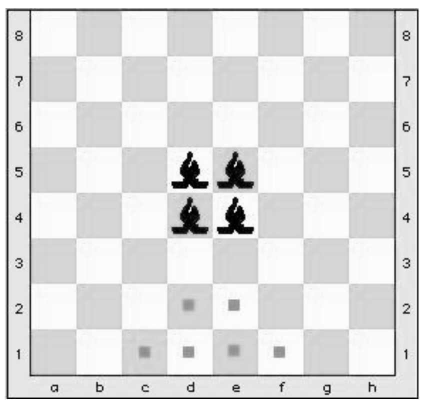

figure 6.1.1b: non-threatened squares

The following algorithm is then used:

- Move the bishop in D5 to C4.
- Move the bishop in E5 to F4.
- If the king is in D1, move the bishop in D4 to C3. Otherwise, move the bishop in E4 to F3.

By first moving the bishop in D5 and E5, the number of non-threatened squares drops to 2. Since the king must be in one of these 2 squares, a move by a bishop that threatens the square he is not in forces the king's next move to be into a threatened square (figure 6.1.1c shows the configuration when the king is in D1). □

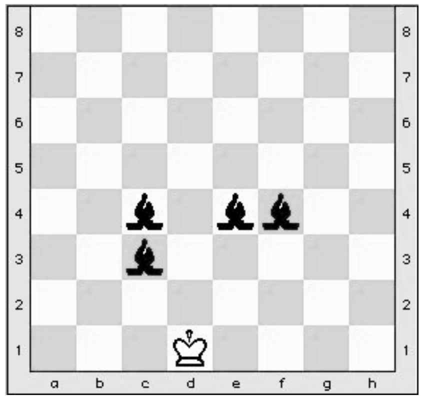

figure 6.1.1c: king in D1 to move into threatened square

Theorem 6.1.2. 2 bishops cannot capture a king.

Proof. To show that 2 bishops are not enough to capture a king, we consider the king in each of the three positions on the board: the middle, an edge, a corner.

- If the king is in a middle square, he has 8 squares into which to move; it is impossible for 2 bishops to threaten these squares.
- If the king is on an edge, he has 5 squares into which to move; these 5 squares cannot be threatened without putting a bishop in danger of capture by the king (see figure 6.1.2).
- If the king is in a corner, he has 3 squares into which to move; likewise, these squares cannot be safely threatened by 2 bishops (see figure 6.1.2).

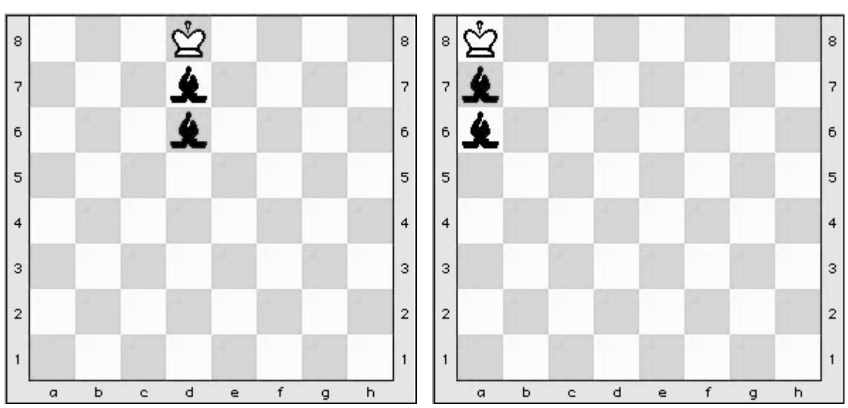

figure 6.1.2: 2 rooks cannot capture a king

6.2. TO CAPTURE A QUEEN WITH BISHOPS

Using the "Attack Everything in Sight" method, 8 bishops are sufficient.

6.3. TO CAPTURE A KNIGHT WITH BISHOPS

Theorem 6.3. 4 bishops can capture a knight.

Proof. The bishops are again placed in the center of the board, as in section 6.1. Without loss of generality, the knight could be in one of 6 free squares, as shown previously in figure 6.1.1b. The following algorithm is then used:

- If the knight starts in C1, D1, or E1, move the bishop in D5 to C4.
- If the knight starts in F1, move the bishop in E5 to F4.
- If the knight starts in D2, move the bishop in D4 to C5, then move the bishop in E5 to F4.
- If the knight starts in E2, move the bishop in E4 to F5, then move the bishop in D5 to C4.

If the knight starts on the edge, all of the squares corresponding to his next move are threatened after one move by the bishops. If, however, the knight starts in either D2 or E2, two moves are needed: the first, to force the knight onto the edge, the second to threaten all possible squares into which the knight could move. □

6.4. TO CAPTURE A ROOK WITH BISHOPS

Using the "Attack Everything in Sight" method, 8 bishops are sufficient.

6.5. TO CAPTURE A BISHOP WITH BISHOPS

Using the "Attack Everything in Sight" method, 8 bishops are sufficient.

BIOGRAPHICAL SKETCH

A senior at Stetson University, Bruce Allan Moser has already proved himself as a talented young pianist. Having performed in North America and Europe, his awards include Stetson University Piano Scholars winner and Winter Park Piano Competition winner.

Bruce's interests branch much further than simply music. He also has majors in both mathematics and German and has been a two-time Georgia state champion in soccer, national champion in the Junior Engineering Technical Society (JETS) competition, a missionary for half year, and has recently returned from studying abroad in Freiburg, Germany for a semester. Bruce resides in Lake Mary, Florida and hope to attend Yale for graduate study in piano.

REFERENCES

[1] Steinhaus, H. *Mathematical Snapshots, 3rd ed.* New York: Dover, pp. 29-30, 1999.

[2] Dudeney, H. E. "The Knight-Guards." §319 in *Amusements in Mathematics.* New York: Dover, p. 95, 1970.

[3] Dudeney, H. E. "Bishops--Unguarded" and "Bishops--Guarded." §297 and 298 in *Amusements in Mathematics.* New York: Dover, pp. 88-89 and 96, 1970.

[4] Fine, Reuben. Basic Chess Endings. Philadelphia: D. McKay Co., 1941.