Rule-Tolerant Verification Algorithms for Completeness of Chinese-Chess Endgame Databases

Haw-ren Fang

Department of Computer Science,
University of Maryland, Maryland, USA
hrfang@cs.umd.edu

Abstract. Retrograde analysis has been successfully applied to solve Awari [6], and construct 6-piece Western chess endgame databases [7]. However, its application to Chinese chess is limited because of the *special rules* about indefinite move sequences. In [4], problems caused by the most influential rule, *checking indefinitely*[1], have been successfully tackled by Fang, with the 50 selected endgame databases were constructed in concord with this rule, where the 60-move rule was ignored. A conjecture is that other special rules have much less effect on staining the endgame databases, so that the corresponding stain rates are zero or small. However, the conjecture has never been verified before. In this paper, a rule-tolerant approach is proposed to verify this conjecture. There are two rule sets of Chinese chess: an Asian rule set and a Chinese rule set. Out of these 50 databases, 24 are verified complete with Asian rule set, whereas 21 are verified complete with Chinese rule set (i.e., not stained by the special rules). The 3 databases, KRKCC, KRKPPP and KRKCGG, are complete with Asian rule set, but stained by Chinese rules.

1 Introduction

Retrograde analysis is widely applied to construct databases of finite, two-player, zero-sum and perfect information games [8]. The classical algorithm first determines all terminal positions, e.g., checkmate or stalemate in both Western chess and Chinese chess, and then iteratively propagates the values back to their predecessors until no propagation is possible. The remaining undetermined positions are then declared as draws in the final phase.

In Western chess, as well as many other games, if a game continues endlessly without reaching a terminal position, the game ends in a draw. However, in Chinese chess, there are special rules other than checkmate and stalemate to end a game. Some positions are to be treated as wins or losses because of these rules, but they are mistakenly marked as draws in the final phase of a typical retrograde algorithm. The most influential special rule is checking indefinitely.

[1] Another name of the concept of checking indefinitely is *perpetual checking.*

H.J. van den Herik et al. (Eds.): CG 2004, LNCS 3846, pp. 129–144, 2006.

In [4], 50 selected endgame databases in concord with this rule were successfully constructed. In this paper, a rule-tolerant verification algorithm is introduced to find out which of these databases are stained by the other special rules.

The organization of this paper is as follows. Section 2 gives the background as the previous works. Section 3 describes the special rules in Chinese chess. Section 4 abstracts these special rules and formulates the problems. Section 5 presents the rule-tolerant algorithms for verifying the completeness of Chinese-chess endgame databases. Section 6 gives the conclusion and suggests two future lines of work. Experimental results are given in the Appendix.

2 Background

Retrograde analysis is applied to the two-player, finite and zero-sum games with perfect information. Such a game can be represented as a *game graph* $G = (V, E)$ which is directed, bipartite, and possibly cyclic, where V is the set of vertices and E is the set of edges. Each *vertex* indicates a position. Each *directed edge* corresponds to a move from one position to another, with the relationship of *parent* and *child* respectively. In Chinese chess, a *position* is an assignment of a subset of pieces to distinct addresses on the board with a certain player-to-move. Positions with out-degree 0 are called *terminal* positions.

2.1 A Typical Retrograde Algorithm

Definition 1. *A win-draw-loss database of a game graph $G = (V, E)$ is a function, $DB : V \to \{\mathbf{win}, \mathbf{draw}, \mathbf{loss}\}$. Each non-terminal position $u \in V$ satisfies the following constraints.*

1. *If $DB(u) = \mathbf{win}$, then $\exists (u, v) \in E$ such that $DB(v) = \mathbf{loss}$.*
2. *If $DB(u) = \mathbf{loss}$, then $\forall (u, v) \in E$, $DB(v) = \mathbf{win}$.*
3. *If $DB(u) = \mathbf{draw}$, then $\exists (u, v) \in E$ such that $DB(v) = \mathbf{draw}$, and $\forall (u, v) \in E$, $(DB(v) = \mathbf{draw}) \vee (DB(v) = \mathbf{win})$.*

Definition 1 draws the most fundamental game-theoretical constraints that a win-draw-loss database must satisfy. A classical retrograde algorithm for constructing a win-draw-loss database consists of three phases: initialization, propagation, and the final phase.

1. In the initialization phase, the win and loss terminal positions are assigned to be **win**s and **loss**es, respectively. They are checkmate or stalemate positions in Chinese chess.
2. In the propagation phase, these values are propagated to the their parents, until no propagation is possible.
3. The final phase is to mark undetermined positions as **draw**s.

In the propagation phase, if an undetermined position has a child being a **loss**, it is assigned as a **win** to satisfy constraint (1) in Definition 1. If an undetermined position have all children as **win**s, it is assigned as a **loss** to satisfy constraint (2). The process continues until no update is possible. The remaining undetermined positions are marked as **draw**s in the final phase. These marked draws satisfy constraint (3). This is a very high level description of retrograde algorithms. In practice, it is usually implemented as: whenever a position has its status determined, it propagates the result to its parents.

Retrograde analysis cannot apply to the whole game graph of Chinese chess, $G = (V, E)$, on a physical computer, because the graph is too big. Therefore, the algorithm is applied to a subgraph $G' = (V', E')$, which satisfies $\forall u \in V'$, $(u, v) \in E \Rightarrow ((v \in V') \wedge ((u, v) \in E'))$. The subgraph is typically partitioned into multiple endgame databases according to the numbers of different pieces remaining on the board. To simplify the notation and without losing generality, this subgraph is also called the Chinese-chess game graph throughout this paper.

2.2 Retrograde Analysis for Chinese Chess

The classical retrograde analysis requires that a game which does not end in a terminal vertex must end in a draw. Otherwise, problems may occur in the final phase. In Western chess, a game which continues endlessly without reaching a terminal vertex is judged to be a draw. In Chinese chess, however, such a game may end in a win or loss. These positions are sometimes mistakenly marked as draws in the final phase. As a result, the draw declaration can only safely be applied to the Chinese-chess endgame databases with only one player having attacking pieces [2,9].

Assuming both players play flawlessly, a position is called *stained* by some special Chinese-chess rule, if the game ends differently (i.e., win-draw-loss status changes) when this rule is ignored. A database is stained if it has one or more stained positions. In contrast, a database is *complete* if all the positions in it have the correct win-draw-loss information. The most influential special rule to stain the endgame databases is checking indefinitely. In [4], 50 endgame databases were successfully constructed in concord with this rule. All the recorded win and loss positions in the databases are verified as correct. However, the marked draws in the databases are possibly stained by other special rules. Therefore, a database is complete, if all the marked draws are not stained. The main theme of this paper is to investigate whether there are positions marked as draws in the databases and whether they are stained by the special rules.

3 Special Rules in Chinese Chess

In Chinese chess, the two sides are called Red and Black. Each side has one King, two Guards, two Ministers, two Rooks, two Knights, two Cannons, and five Pawns, which are abbreviated as K, G, M, R, N, C and P, respectively[2]. The pieces Rooks, Knights, Cannons and Pawns are called *attacking pieces* since they

[2] The English translation of the Chinese names differs by author.

can move across the *river*, the imaginary stream between the two central horizontal lines of the board. In contrast, Guards and Ministers are called *defending* pieces because they are confined to the domestic region[3].

In addition to checkmate and stalemate, there are various rules of indefinite move sequences to end a game. They are called *special rules* in this paper. An indefinite move sequence is conceptually an infinite move sequence. In real games, it is determined by the threefold repetition of positions in a finite move sequence [1, page 20, rule 23, page 65, rule 3].

3.1 Chinese and Asian Rule Sets

There are two rule sets of Chinese chess: an Asian rule set and a Chinese rule set. The differences are generally about the special rules other than checking indefinitely. In both Asian and Chinese rule sets, there are dozens of detailed special rules and sub-rules. Some rules are exceptions of some others. Because they are very complicated, we attempt to verify the completeness of a given endgame database via a *rule-tolerant* approach, instead of formulating all these rules.

All the special rules discussed in this paper refer to the rule book [1], in which pages 1–46 describe the Chinese rule set, and pages 47–119 describe the Asian rule set. Readers do not require this book to follow this paper. However, this rule book keeps to be cited to confirm that the theory is correct. The rules of checking indefinitely and mutual checking indefinitely are well studied in [3,4]. We focus on the other special rules.

3.2 The Rules of Chasing Indefinitely

In Chinese chess, *chasing indefinitely* is forbidden. The general concept is that a player cannot chase some opponent's piece continuously without ending [1, page 21, page 64]. The term *chase* is defined similarly to the term check, but the prospective piece to be captured is not the King but some other piece. For example in Figure 1(a) with Red to move, the game continues cyclically with moves Re0-e2 Ng2-f0 Re2-e0 Nf0-g2, etc. Red loses the game since he[4] is *forced* to chase the Black Knight endlessly. Here and throughout this paper, a player is said to be forced to play the indicated moves, if he will lose the game by making any other moves because of the rules of checkmate, stalemate or checking indefinitely.

In some cases, chasing is allowed. For example, the Kings and the Pawns are allowed to chase other pieces [1, page 22, rule 27, page 65, rule 9]. In Figure 1(b) with Red to move, the game continues cyclically with moves Rb0-c0 Pb1-c1 Rc0-b0 Pc1-b1, etc. Although Black chases the Red Rook endlessly, the game ends in a draw because the chaser is a Pawn. Besides, some types of chasing

[3] The notation and basic rules of Chinese chess in English can be found in the ICGA web page `http://www.cs.unimaas.nl/icga/games/chinesechess/`, and in FAQ of the Internet news group `rec.games.chinese-chess`, which is available at `http://www.chessvariants.com/chinfaq.html`.

[4] In this paper we use 'he' when 'he' and 'she' are both possible.

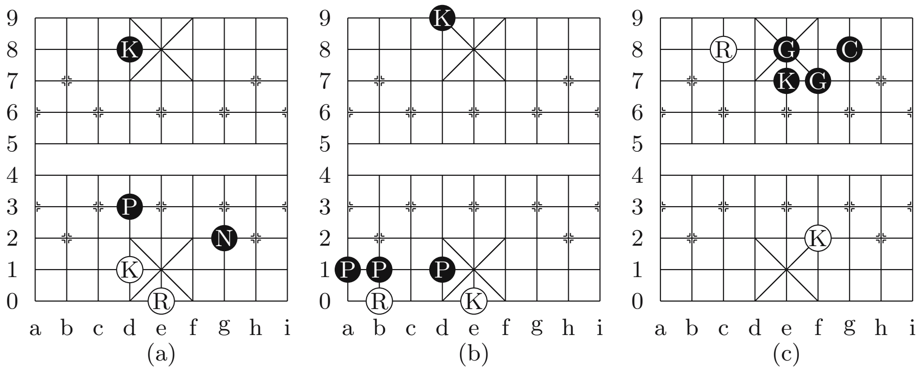

Fig. 1. Examples to illustrate the special rules

indefinitely is allowed under the Asian rule set, but forbidden by the Chinese rules. For example, it is allowed in the Asian rule set to endlessly chase one piece every other move and chase another piece at the moves in between [1, page 103, rule 32], whereas it is forbidden by the Chinese rules [1, page 24, rule 28.12]. An example is given in Figure 2(c). The rules of chasing indefinitely may also stain the endgame databases.

3.3 Summary of Special Rules

An indefinite move sequence is composed of two *semi-sequences*: one consists of the moves by Red and the other has the moves by Black. A semi-sequence is classified as being *allowed* or *forbidden* by the special rules. In the 50 endgame databases in concord with checking indefinitely, only marked draws are possibly stained. Therefore, special rules about allowed semi-sequences of moves can be ignored. In the Asian rule set, a forbidden semi-sequence of moves is either checking indefinitely or chasing indefinitely [1, page 64–65]. In the Chinese rule set, a forbidden semi-sequence of moves consists of three types of moves: checking, chasing, and *threatening to checkmate* [1, page 21]. A threatening-to-checkmate move means that a player *attempts* to checkmate his opponent by a semi-sequence of checking moves. A semi-sequence of moves is *allowed* if it is not forbidden. According to [1, page 20, rule 24, page 64, section 2], the special rules applied to an indefinite move sequence are summarized as follows.

1. If only one player checks the other indefinitely, the player who checks loses the game.
2. Otherwise, if only one semi-sequence of moves is forbidden and the other is allowed, the player who played the forbidden semi-sequence of moves loses.
3. Otherwise, the game ends in a draw.

With the above summary, both mutual checking indefinitely and mutual chasing indefinitely result in a draw. The example in Figure 1(c) with Red to move illustrates the difference between the Asian and the Chinese rule sets. The game continues cyclically with moves Rc8-c7 Ge8-d7 Rc7-c8 Gd7-d8, etc. Red is forced to check every other move and chases at the moves in between. It is allowed by the Asian rules [1, page 64, rule 2] but forbidden by the Chinese rules [1, page 21, rule 25.2].

4 Problem Formulation

Below we formulate the problem to be solved more precisely. In 4.1 we deal with abstracting the special rules and in 4.2 we describe a rule-tolerant approach.

4.1 Abstracting Special Rules

Denote the Chinese-chess game graph by $G = (V, E)$. For ease of discussion, we assume V contains the illegal positions, in which the own King is left in check, and E includes the moves to these illegal positions. Given a position $v \in V$, we define $\overline{v} \in V$ to have the piece assignment on the board the same as that of v but with a different player-to-move. The boolean function $check : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ is to indicate whether the edge (u, v) is a checking move. In other words, $check((u, v)) = \mathbf{true}$ if and only if the next mover in $\overline{v}$ can capture the opponent's King immediately.

Definition 2. *In an infinite sequence of moves (v_0, v_1), (v_1, v_2), etc., the first mover loses the game because of the rule of checking indefinitely, if*

1. *$\forall$ even i, $check((v_i, v_{i+1})) = \mathbf{true}$.*
2. *$\forall n \geq 0$, $\exists$ odd $j > n$, such that $check((v_j, v_{j+1})) = \mathbf{false}$.*

In addition, the game results in a draw because of mutual checking indefinitely if, $\forall i \in N \cup \{0\}$, $check((v_i, v_{i+1})) = \mathbf{true}$.

Note that this definition ignores the 60-move rule. Two semi-sequences of indefinite moves are (v_0, v_1), (v_2, v_3), etc. and (v_1, v_2), (v_3, v_4), etc.

We use the boolean function $threaten : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ to indicate whether a given move is threatening to checkmate. It is defined by $threaten((u, v)) = \mathbf{true}$ if the own King in u is not in check and the next mover in $\overline{v}$ can checkmate his opponent with a semi-sequence of checking moves, assuming both players play flawlessly. If the next mover in u is in check, the move (u, v) is generally to get out of check and not counted as an attempt to checkmate the opponent.

Before defining a chasing move, we need to define the capturing move. A *direct capturing* move has a capturing piece and a captured piece other than the King[5]. Two direct capturing moves are treated the same if they have the same capturing piece and captured piece. A move (u, v) is called a chasing move, if $\overline{v}$ has a capturing move which u does not have and after the capturing move from $\overline{v}$, the own King of v is not in check (i.e., not an illegal position) [1, page 24, rule 29]. The boolean function $chase^* : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ is defined to indicate

[5] The cases of capturing a protected piece and indirect capturing are omitted here but will be discussed in Subsection 4.2.

whether a given edge (u, v) is a chasing move. The boolean function $chase: E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ is to indicate whether a given edge is a forbidden chasing move. Note that $(chase((u, v)) = \mathbf{true}) \Rightarrow (chase^*((u, v)) = \mathbf{true})$. The definition of mutual and non-mutual chasing indefinitely is similar to that of mutual and non-mutual checking indefinitely in Definition 2, but replacing $check((u, v))$ by $chase((u, v))$. For the Asian rule set, we assume that the given database to be verified is in concord with checking indefinitely and mutual checking indefinitely, such as the 50 endgame databases in [4]. Therefore, we may focus only on chasing indefinitely in the verification algorithms for the Asian rules.

Definition 3. *Let the boolean function* $forbid : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ *indicate whether a given move is forbidden. Then,*

1. *Asian rules:* $forbid((u, v)) := chase((u, v))$.
2. *Chinese rules:* $forbid((u, v)) := check((u, v)) \vee threaten((u, v)) \vee chase((u, v))$.

If a semi-sequence of indefinite moves (v_0, v_1), (v_2, v_3), *etc. is forbidden, then* $\forall$ *even* i, $forbid((v_i, v_{i+1})) = \mathbf{true}$.

4.2 A Rule-Tolerant Approach

There is a problem we need to face. The value of $chase((u, v))$ may depend on the other moves in a move sequence. Sometimes we cannot determine $chase((u, v))$ without inspecting the other moves in the move sequence. We call such a move *path-dependent*. For example, usually Pawns are allowed to chase; therefore, $chase((u, v)) = \mathbf{false}$ if the chasing piece is a Pawn. However, with the Chinese rule set, if the opponent plays a forbidden semi-sequence of moves at the same time, then $chase((u, v)) = \mathbf{true}$ with the piece to chase being a Pawn [1, page 32, rule 11]. Another example in the Asian rule set is: indefinitely chasing one piece every other move and chasing another piece at the moves in between is allowed [1, page 103, rule 32]. As a result, $forbid((u, v))$ is path-dependent. Nevertheless, $check((u, v))$, $threaten((u, v))$, $chase^*((u, v))$ and $forbid^*((u, v))$ are path-independent.

A rule-tolerant approach is proposed as follows. Let the boolean function $f(s)$ indicate whether a given semi-sequence of moves s is forbidden. Instead of programming the function $f(s)$, we look for another boolean function $f^*(s)$ satisfying $(f(s) = \mathbf{true}) \Rightarrow (f^*(s) = \mathbf{true})$. In other words, if we know $f^*(s) = \mathbf{false}$, we can safely declare s is not a forbidden semi-sequence of moves. If a move sequence is composed of two semi-sequence of moves satisfying $f^*(s) = \mathbf{false}$, the game is verified as a draw. If all positions marked as draws in a given database DB are verified as draws, then the database can be declared complete, assuming only marked draws are possibly stained.

Lemma 1. *Define the boolean function* $forbid^* : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ *as follows.*

1. *Asian rules:* $forbid^*((u, v)) := chase^*((u, v))$.
2. *Chinese rules:* $forbid^*((u, v)) := check((u, v)) \vee threaten((u, v)) \vee chase^*((u, v))$.

A semi-sequence of moves (v_0, v_1), (v_2, v_3), *etc. is called suspiciously forbidden if,* $forbid^*((v_i, v_{i+1})) = \mathbf{true}$ *for all even* i. *If a semi-sequence of moves is not suspiciously forbidden, it is an allowed semi-sequence of indefinite moves.*

Proof. By Definition 3 and $(chase((u, v)) = \mathbf{true}) \Rightarrow (chase^*((u, v)) = \mathbf{true})$, $(forbid((u, v)) = \mathbf{true}) \Rightarrow (forbid^*((u, v)) = \mathbf{true})$. A forbidden semi-sequence of moves (v_0, v_1), (v_2, v_3), etc. satisfies that $\forall$ even j, $forbid((v_j, v_{j+1})) = \mathbf{true}$. If it is not suspiciously forbidden, then $\exists$ even i such that $forbid^*((v_i, v_{i+1})) = \mathbf{false}$, which implies $forbid((v_i, v_{i+1})) = \mathbf{false}$. A contradiction. □

In both Chinese and Asian rule sets, capturing a *protected* piece is usually considered as not a capturing move [1, page 24, rule 29.3, page 108, rule 35]. We call them *nullified* capturing moves in this paper. The general concept of a protected piece p is that, if p is captured by some opponent's piece q, the player of p can capture q back immediately *without exposing* the own King in check [1, page 24, rule 29.4, page 107, rule 34]. It has an exception in the Asian rule set: chasing a protected Rook by Knights and Cannons is forbidden [1, page 86, rule 20]. It is also true in the Chinese rule set because the chaser attempts to gain from capturing [1, page 21, rule 25.3, page 24, rule 29.1]. The key for being rule-tolerant is $(chase((u, v)) = \mathbf{true}) \Rightarrow (chase^*((u, v)) = \mathbf{true})$. The function $chase^*((u, v))$ is defined by the chasing moves of u and $\overline{v}$. Ignoring nullified capturing moves of $\overline{v}$ remains rule-tolerant, because $chase^*((u, v))$ has a greater chance to be **true**. The problem occurs only when u has a nullified capturing move, which remains a capturing move but not nullified in $\overline{v}$. Figure 2(c) gives such an example. In the Appendix, nullified capturing moves are taken into account in the experiments. It is also shown that ignoring this rule in the verification algorithms does not change the conclusion from the experiments on the 50 endgame databases.

All the Asian rules are taken into account in the rule-tolerant approach, but we do ignore two Chinese rules about forbidden moves in the current experiments. (a) One is *indirect capturing*. It means that a move does not capture a piece immediately, but will capture some opponent's piece after a semi-sequence of checking moves made by the capturing player [1, page 24, rule 29.1]. (b) The other is capturing an *insufficiently* protected piece. It means that the player can *gain* after a sequence of capturing moves by both players [1, page 24, rule 29.3]. It is an exception of chasing a protected piece, i.e., capturing a protected piece is usually not counted as a capturing move, unless the capturing player can gain some piece after a sequence of capturing moves. The function $chase^*((u, v))$ is defined by the chasing moves of u and $\overline{v}$. With the Chinese rule set, problems may occur when $\overline{v}$ has a capturing move of case (a) or (b) which u does not have. These two rules require substantial programming. For case (a), all the finite semi-sequences of checking moves ending with capturing moves need to be considered. For case (b), the order of the capturing needs to be taken into account. Cases mixed with (a) and (b) are even more complicated.

If a player can win the game by either checkmate or stalemate, the position is already correctly marked by a typical retrograde algorithm. If a player can win the game by checkmate, stalemate, or the rule of checking indefinitely, then the position is already correctly marked by a checking-indefinitely-concordant retrograde algorithm by Fang [4]. Given a database from [4], our consideration deals with the positions in which some player can win the game only by special rules other than checking indefinitely, assuming both players play flawlessly. The experiments are on the 50 endgame databases in [4]; each has at most three attacking pieces (except KRKPPP which has four) remaining on the board and a database size less than 1GB. Usually, it requires several attacking pieces to form an indefinite move sequence of cases (a) or (b). Therefore, the effect because of ignoring the two Chinese rules in the experiments tends to be very minor.

5 Rule-Tolerant Verification Algorithms

Given a checking indefinitely concordant database of Chinese chess, in which the marked win and loss positions have correct information, we need to foresee if any of the marked draws may be mistaken because of the special rules. To achieve this goal, *suspicious move patterns* of special rules are defined (in 5.1), computed (in 5.2), and verified (in 5.3).

5.1 Suspicious Move Patterns of Special Rules

For the ease of discussion, we assume there is an *attacking* side and the other side is *defending*. The attacking side tries to win the game by forcing the defending side play the forbidden semi-sequence of moves, whereas the defending side tries to avoid such a semi-sequence.

Definition 4. *Given a win-draw-loss database DB of the Chinese-chess game graph $G = (V, E)$, a suspicious move pattern of special rules is a subgraph of G, denoted by $G^* = (V^*, E^*)$, with V^* being partitioned into V_A^* and V_D^* according to whether the next mover is attacking or defending. G^* satisfies the following constraints.*

1. $\forall u \in V^*$, $DB(u) = \mathbf{draw}$.
2. $\forall (u, v) \in E^*$ *with* $u \in V_D^*$, $forbid^*((u, v)) = \mathbf{true}$.
3. $\forall u \in V_D^*$, $((u, v) \in E) \wedge (DB(v) = \mathbf{draw}) \Rightarrow ((u, v) \in E^*)$.
4. $\forall u \in V^*$, $\exists (u, v) \in E^*$, *i.e., out-degree is at least* 1.

Constraint (1) is because only positions marked as draws are possibly stained. Constraint (2) ensures the defending side play suspicious forbidden moves all the time inside the pattern. Constraint (3) makes the defending side unable to quit the pattern without losing the game. Constraint (4) keeps the pattern indefinite.

Lemma 2. *Given a win-draw-loss database for the Chinese-chess game graph G, and any two suspicious move patterns of special rules $G_1^* = (V_1^*, E_1^*)$ and $G_2^* = (V_2^*, E_2^*)$ having the same attacking and defending sides, $G^* = (V_1^* \bigcup V_2^*, E_1^* \bigcup E_2^*)$ is also a suspicious move pattern of special rules.*

Proof. The graph G^* clearly satisfies constraints (1) and (4). $\forall u \in V_1^*$ (or V_2^*), if the next mover in u is defending, no more edges going out of u are added in the union. Therefore, constraints (2) and (3) are satisfied. □

Theorem 1. *Given a win-draw-loss database of Chinese chess, there exist two unique maximum suspicious move patterns of special rules. In one of them, Red is the attacking side, whereas in the other, Black is the attacking side.*

Proof. Since the game graph $G = (V, E)$ of Chinese chess is finite, the number of suspicious move patterns of special rules is finite, with a given win-draw-loss database of Chinese chess. Denote these move patterns with Red as attacking side by $G_i^* = (V_i^*, E_i^*)$ for $i = 1, 2, \ldots, n$. By Lemma 2, $\overline{G^*} = (\bigcup_{i=1}^n V_i^*, \bigcup_{i=1}^n E_i^*)$ is a suspicious move pattern of special rules. It is maximum since $V_j^* \subseteq \bigcup_{i=1}^n V_i^*$ and $E_j^* \subseteq \bigcup_{i=1}^n E_i^*$ for $j = 1, 2, \ldots, n$. The proof is completed by swapping the attacking and defending sides. □

Given a win-draw-loss database of Chinese chess in concord with checking indefinitely from [4], each forbidden semi-sequence of indefinite moves is inside one of the two maximum patterns, assuming that both players, based on the given win-draw-loss database, play flawlessly to avoid playing and force each other to play a suspiciously forbidden semi-sequence of indefinite moves. By Lemma 1, a forbidden semi-sequence of moves must be suspiciously forbidden. Therefore, if these two maximum suspicious move patterns of special rules are empty, then the given database is complete. The discussion proves the following theorem.

Theorem 2. *Given a win-draw-loss database DB of Chinese chess in concord with checking indefinitely, if both maximum suspicious move patterns of special rules are empty, then database DB is complete.*

5.2 Computing the Maximum Suspicious Move Patterns

Lemma 3. *Given a win-draw-loss database for the Chinese-chess game graph* $G = (V, E)$*, the two maximum suspicious move patterns of special rules* $\overline{G^*} = (\overline{V^*}, \overline{E^*})$ *are induced subgraphs of G, i.e.,* $\forall (u, v) \in E$, $(u, v \in \overline{V^*}) \Rightarrow ((u, v) \in \overline{E^*})$.

Proof. Given $(u, v) \in E$, if the next mover of u is the defending side, the statement is true by constraints (1) and (3) in Definition 4. If the next mover of u is the attacking side, the statement is true because the graph $\overline{G^*}$ is maximum. □

By Lemma 3, if we know $\overline{V^*}$ with a given win-draw-loss database DB, then $\overline{G^*} = (\overline{V^*}, \overline{E^*})$ can be determined as an induced subgraph of $G = (V, E)$. Define $\overline{V} = \{u : (u \in V) \wedge (DB(u) = \textbf{draw})\}$ and $\overline{G} = (\overline{V}, \overline{E})$ as an induced subgraph of G, i.e., $\overline{E} := \{(u, v) : ((u, v) \in E) \wedge (u, v \in \overline{V})\}$. Then $\overline{G^*}$ is a subgraph of $\overline{G}$ because of constraint (1) in Definition 4. The algorithm to compute $\overline{V^*}$ consists of two phases: initialization and pruning. The first phase is to compute suspicious win and loss candidate sets W and L respectively, such that $\overline{V^*} \subset W \bigcup L$. The pseudo-code of initialization phase is as follows.

$W \leftarrow \emptyset$, $L \leftarrow \emptyset$
for all $u \in \overline{V}$ with the next mover of u as the defending side **do**
  if $\forall(u,v) \in \overline{E}$, $forbid^*((u,v)) =$ **true then**
    $L \leftarrow L \bigcup \{u\}$
    $\forall(w,u) \in \overline{E}$, $W \leftarrow W \bigcup \{w\}$
  end if
end for

The second phase is to prune unqualified candidates in W and L by an iterative process, until no pruning is possible. If a suspicious loss candidate $u \in L$ has a child $v \in \overline{V}$ but $v \notin W$, then u does not satisfy constraint (3) in Definition 4 and therefore is pruned. If a suspicious win candidate in W does not have a child in L, then it does not satisfy constraint (4) and therefore is pruned. This observation suggests the following algorithm.

repeat
  {Prune unqualified suspicious loss positions.}
  for all $u \in L$ **do**
    if $\exists(u,v) \in \overline{E}$ such that $v \notin W$ **then**
      $L \leftarrow L - \{u\}$
    end if
  end for
  {Prune unqualified suspicious win positions.}
  for all $v \in W$ **do**
    if $\forall(v,u) \in \overline{E}$, $u \notin L$ **then**
      $W \leftarrow W - \{v\}$
    end if
  end for
until No more pruning is possible.

When no more pruning is possible, the subgraph of G induced by $W \bigcup L$ satisfies all constraints in Definition 4. Therefore, it is the maximum suspicious move pattern of special rules $\overline{G^*} = (\overline{V^*}, \overline{E^*})$, i.e., $\overline{V^*} = W \bigcup L$. Swapping the attacking and defending sides, we obtain the other maximum suspicious move pattern. Some strategies, such as children counting, can be applied to improve efficiency.

5.3 Verification Algorithms

By Theorem 2, if both the maximum suspicious move patterns of special rules are empty, then the given database DB is complete. This verification algorithm is called rule-tolerant because of the tolerance between boolean functions $chase^*((u,v))$ and $chase((u,v))$. If these two maximum suspicious move patterns are small enough, we may inspect the positions inside the patterns with a rule book to see if they are truly stained by the special rules. Besides, we may try trimming the maximum move patterns by reducing the tolerance.

Lemma 4. *Assume we are given a path-independent boolean function* $chase^{\overline{*}}$: $E \rightarrow \{\textbf{true}, \textbf{false}\}$, *which satisfies* $(chase((u,v)) = \textbf{true}) \Rightarrow (chase^{\overline{*}}((u,v)) = \textbf{true}) \Rightarrow (chase^*((u,v)) = \textbf{true})$ *for all* $(u,v) \in E$, *where* $G = (V,E)$ *is* *the Chinese-chess game graph. Define the boolean function* $\overline{forbid^*} : E \rightarrow \{\mathbf{true}, \mathbf{false}\}$ *based on* $\overline{chase^*}$ *in a similar way to* $forbid^*(u, v)$ *in Lemma 1. The resulting maximum suspicious move pattern* $\overline{G^*}$ *is a subgraph of* G^* *in Theorem 1, with given the same win-draw-loss database. All forbidden indefinite semi-sequences of moves are inside* $\overline{G^*}$*, with the assumption that both players play flawlessly.*

Proof. For all $(u, v) \in E$, $(\overline{forbid^*}((u, v)) = \mathbf{true}) \Rightarrow (forbid^*((u, v)) = \mathbf{true})$, because $(\overline{chase^*}((u, v)) = \mathbf{true}) \Rightarrow (chase^*((u, v)) = \mathbf{true})$. Therefore, a suspicious move pattern based on $\overline{chase^*}$ is also suspicious based on $chase^*$. The resulting maximum suspicious move pattern $\overline{G^*}$ based on $\overline{chase^*}$ is a subgraph of G^* based on $chase^*$. With the fact $(chase((u, v)) = \mathbf{true}) \Rightarrow (\overline{chase^*}((u, v)) = \mathbf{true})$ for all $(u, v) \in E$, the last statement is true by the discussion similar to the proof of Lemma 1 and Theorem 2. □

The function $\overline{chase^*}((u, v))$ is obtained from $chase^*((u, v))$ by excluding some path-independent allowed moves. The algorithm in Subsection 5.2 requires the boolean function $forbid^*((u, v))$ being path-independent. Therefore, we need to keep function $\overline{chase^*}((u, v))$ being path-independent when reducing the tolerance. The closer $\overline{chase^*}((u, v))$ to $chase((u, v))$, the smaller the tolerance, and the smaller the maximum suspicious move patterns we may obtain.

Theorem 3. *Given a win-draw-loss database with marked draws possibly stained and the attacking side specified, the maximum suspicious move pattern of special rules with the Asian rule set is a subgraph of that with the Chinese rule set, assuming they use the same function* $chase^*((u, v))$.

Proof. By the definition of $forbid^*((u, v))$ in Lemma 1, a suspiciously forbidden semi-sequence of moves with the Asian rule set is also suspiciously forbidden with the Chinese rule set. Therefore, a suspicious move pattern of special rules with the Asian rule set is also suspicious with the Chinese rule set, which implies this theorem. □

By Theorem 3, if both maximum suspicious move patterns are empty with the Chinese rule set, then they are also empty with the Asian rule set, and therefore the given win-draw-loss database is complete with both the Asian and Chinese rule sets.

6 Conclusion and Future Work

Retrograde analysis has been successfully applied to many games. In Chinese chess, its application is confined to the endgames with only one player having attacking pieces. Other endgame databases were not perfectly reliable because of the existence of special rules. In the experiments with the 50 endgame databases in concord with checking indefinitely in [4], 24 and 21 endgame databases with both players having attacking pieces are verified complete with the Asian and Chinese rule sets, respectively. The 3 endgame databases, KRKCC, KRKPPP and KRKCGG, are complete with the Asian rule set but stained by the Chinese rules, as shown in the Appendix. Two suggested future directions of work are listed below.

1. **Knowledgeable encoding and querying of endgame databases.** For the endgame databases verified to be complete, we may extract and condense the win-draw-loss information into physical memory via the approach by Heinz [5]. The result can improve the present Chinese-chess programs.
2. **Construction of complete endgame databases.** For those stained by the special rules other than checking indefinitely, further work is required for the Chinese-chess endgame databases with complete information.

Acknowledgement

The author would like to thank Tsan-sheng Hsu for his suggestion to tackle this problem, and Shun-Chin Hsu for sharing his experience of programming threatening to checkmate.

References

1. China Xiangqi Association. *The Playing Rules of Chinese Chess*. Shanghai Lexicon Publishing Company, 1999. In Chinese.
2. H.-r. Fang, T.-s. Hsu, and S.-c. Hsu. Construction of Chinese Chess endgame databases by retrograde analysis. In T. Marsland and I. Frank, editors, *Lecture Notes in Computer Science 2063: Proceedings of the 2nd International Conference on Computers and Games*, pages 96–114. Springer-Verlag, New York, NY, 2000.
3. H.-r. Fang, T.-s. Hsu, and S.-c. Hsu. Indefinite sequence of moves in Chinese Chess endgames. In J. Schaeffer, M. Müller, and Y. Björnsson, editors, *Lecture Notes in Computer Science 2063: Proceedings of the 3rd International Conference on Computers and Games*, pages 264–279. Springer-Verlag, New York, NY, 2002.
4. H.-r. Fang, T.-s. Hsu, and S.-c. Hsu. Checking indefinitely in Chinese-chess endgames. *ICCA Journal*, 27(1):19–37, 2004.
5. E. A. Heinz. Knowledgeable encoding and querying of endgame databases. *ICCA Journal*, 22(2):81–97, 1999.
6. J.W. Romein and H.E. Bal. Awari is solved. *ICCA Journal*, 25(3):162–165, 2002.
7. K. Thompson. 6-piece endgames. *ICCA Journal*, 19(4):215–226, 1996.
8. H. J. van den Herik, J. W. H. M. Uiterwijk, and J. van Rijswijck. Games solved: Now and in the future. *Artificial Intelligence*, 134:277–311, 2002.
9. R. Wu and D.F. Beal. Fast, memory-efficient retrograde algorithms. *ICCA Journal*, 24(3):147–159, 2001.

Appendix: Experimental Results on the 50 Endgame Databases

Before verifying the completeness of a given endgame database as the Chinese-chess game graph is split, we need to verify all its supporting databases. If any of the two maximum suspicious move patterns of the special rules are non-empty, we may inspect whether these suspicious positions are stained by the special rules. An endgame database is not complete, if it has positions truly stained. If both maximum suspicious move patterns of the special rules of a given endgame database are empty or all the suspicious positions are inspected to be not stained, then the endgame database is verified as complete, assuming all its supporting databases are complete. The overall procedure to verify a given endgame database includes verifying all its supporting databases in bottom-up order.

The experiments are performed on the 50 endgame databases in concord with checking indefinitely in [4]. Table 1 lists the statistics for the Asian rule set. It provides the number of positions for each maximum suspicious move patterns. There

Table 1. Statistics of max suspicious patterns of 50 endgame databases, Asian rules

Database Name	*Comp-lete*	*Number of Positions* legal	draw	*Ignore Protecting* Red attack	Black attack	*Consider Protecting* Red attack	Black attack
KRCKRGG	N	252077421	147261943	6152	517	5573	170
KRCKRG	N	141001563	66005451	90775	54	88632	156
KRCKRM	N	209431807	103902489	22361	974	21948	1214
KRCGGKR	N	256862617	34156099	271110	23970	269839	23970
KRCGKR	N	142232812	14112951	54040	354	54324	354
KRCMKR	N	210030190	23470723	58314	8865	29958	8865
KRCKR	N	31012335	9108171	4132	0	3917	0
KRNKRGG	N	251181481	103598842	20739	2428	16096	2196
KRNKRG	N	138660209	15453492	7877	977	7356	1142
KRNKRM	N	204263530	29396541	11220	2254	9342	2915
KRNKR	N	30118362	2936309	453	192	449	192
KRPKRG	N	84330363	35495583	31558	58	25360	187
KRPKRM	N	124050578	44636393	23584	1218	18256	1334
KRPKR	N	18443469	2635006	975	0	930	0
KRKNN	N	16300026	4621246	357	0	363	344
KRKNC	N	33568194	596093	296	162	296	162
KRKCC	Y	17300976	1409308	0	0	0	0
KRKNPGG	N	168307887	2123952	427	382	427	382
KRKNPG	N	92456806	112434	0	237	0	237
KRKNPM	N	136200539	466627	0	492	0	492
KRKNP	N	20011890	20026	0	79	0	79
KRKPPP	Y*	103676439	1179271	111	72	111	24
KRKPPGG	Y*	52598998	947571	0	61	0	61
KRKPPMM	Y	122221940	2211873	0	0	0	0
KRKPPG	Y*	28498574	53263	0	26	0	26
KRKPPM	Y	41658907	107613	0	0	0	0
KRKPP	Y	6084903	8187	0	0	0	0
KNPKN	N	21682338	4889447	17760	0	17746	0
KNPKCG	N	100076040	40930099	1917	0	1899	0
KNPKCM	N	149719630	67281695	7970	0	7142	0
KNPKC	N	22364304	6387677	268	0	301	0
KCPKC	Y	22956705	20094132	0	0	0	0
KRGGKR	Y	2997932	1840528	0	0	0	0
KRGKR	Y	1628603	966336	0	0	0	0
KRMKR	Y*	2389472	1370793	29	144	29	144
KRKR	Y	348210	193950	0	0	0	0
KRKNGGMM	N	63684381	33604025	83	0	83	0
KRKNGGM	Y	21879507	719210	0	0	0	0
KRKNGMM	N	35195142	4703606	1605	0	1622	0
KRKNGG	Y	3221138	6498	0	0	0	0
KRKNGM	Y	12032732	27288	0	0	0	0
KRKNMM	Y	7654095	121216	0	0	0	0
KRKNG	Y	1762807	3275	0	0	0	0
KRKNM	Y	2605497	5200	0	0	0	0
KRKN	Y	380325	609	0	0	0	0
KRKCGG	Y	3322727	1379102	0	0	0	0
KRKCMM	Y	7913097	2969808	0	0	0	0
KRKCG	Y	1820350	2462	0	0	0	0
KRKCM	Y	2694400	91748	0	0	0	0
KRKC	Y	393327	7479	0	0	0	0

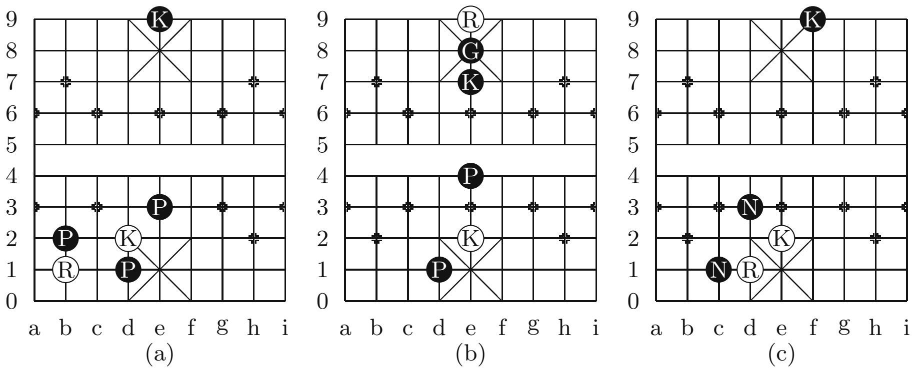

Fig. 2. (a)(b) Suspicious positions not stained by the special rules. (c) A game ends differently under the Asian and Chinese rule sets.

are two experiments on each database. One ignores the rule of capturing a protected piece discussed in Subsection 4.2, and the other takes it into account. Although the statistics differ, the conclusion does not change with this rule ignored. An endgame database with a non-empty maximum suspicious move pattern but verifiedas complete is denoted by Y*. The statistics exclude illegal positions. Two conjugate positions (i.e., piece assignment on the board is the same as each other in the mirror) are treated as the same one and counted only once.

In Figure 2(a), the game continues cyclically with moves Rb1-a1 Pb2-a2 Ra1-b1 Pa2-b2, etc. In Figure 2(b), the game continues indefinitely with moves Ke2-d2 Pd1-e1 Kd2-e2 Pe1-f1 Ke2-f2 Pf1-e1, etc. In both games, Red is forced to chase indefinitely. However, both are allowed because the chaser is a King or a Pawn [1, page 22, rule 27, page 65, rule 9]. In Figure 2(c), the game continues indefinitely with moves Ke2-d2 Nc1-b3 Kd2-e2 Nb3-c1, etc. Red is forced to chase the two Black Knights iteratively. It is allowed under the Asian rule set but forbidden by the Chinese rules [1, page 24, rule 28.12, page 103, rule 32].

A forbidden semi-sequence of moves under the Asian rule set is generally also forbidden under the Chinese rule set. To verify the completeness in Chinese rules, we may focus on the endgame databases complete with the Asian rule set. Table 2 lists the statistics of the experimental results for the Chinese rule set. It includes all the endgame databases complete with the Asian rule set. An endgame database complete with the Asian rule set but stained by the Chinese rules is denoted by N*. Examples are KRKCC, KRKPPP, and KRKCGG. Figure 3(a) illustrates a KRKCC endgame stained only by the Chinese rules. The game continues cyclically with the moves Ke2-f2 Ce6-f6 Kf2-e2 Cf6-e6, etc. Black is forced to check every other move, and threatens to checkmate at all moves in between. In Figure 3(b), the game continues indefinitely with the moves Rb8-c8 Pc1-b1 Rc8-c9 Kf9-f8 Rc9-b9 Pb1-c1, etc. Red is forced to play a semi-sequence of indefinite moves which consists of checking and chasing moves. In Subsection 3.3, Figure 1(c) illustrates another example of a KRKCGG endgame stained only by the Chinese rules.

In Chinese chess, Cannon requires an additional piece to jump over to capture. In KNPKC endgames, Red seems unlikely to force Black to play forbidden chas-

Table 2. Statistics of max suspicious patterns of 35 endgame databases, Chinese rules

Database Name	*Comp-lete*	*Number of Positions*		*Ignore Protecting*		*Consider Protecting*	
		legal	*draw*	*Red attack*	*Black attack*	*Red attack*	*Black attack*
KRKNN	N	16300026	4621246	372	76	378	344
KRKCC	N*	17300976	1409308	602	0	602	0
KRKNPGG	N	168307887	2123952	1187	382	1188	382
KRKNPG	N	92456806	112434	348	237	348	237
KRKNPM	N	136200539	466627	725	519	725	519
KRKNP	N	20011890	20026	116	79	116	79
KRKPPP	N*	103676439	1179271	129	781	129	268
KRKPPGG	Y*	52598998	947571	0	68	0	68
KRKPPMM	Y	122221940	2211873	0	0	0	0
KRKPPG	Y*	28498574	53263	0	35	0	35
KRKPPM	Y	41658907	107613	0	0	0	0
KRKPP	Y	6084903	8187	0	0	0	0
KNPKN	N	21682338	4889447	18116	0	18092	0
KNPKCG	N	100076040	40930099	12979	0	11227	0
KNPKCM	N	149719630	67281695	10407	0	9410	0
KNPKC	N	22364304	6387677	2838	0	1060	0
KCPKC	Y	22956705	20094132	0	0	0	0
KRGGKR	Y	2997932	1840528	0	0	0	0
KRGKR	Y	1628603	966336	0	0	0	0
KRMKR	Y*	2389472	1370793	29	144	29	144
KRKR	Y	348210	193950	0	0	0	0
KRKNGGMM	N	63684381	33604025	83	0	83	0
KRKNGGM	Y	21879507	719210	0	0	0	0
KRKNGMM	N	35195142	4703606	1687	0	1708	0
KRKNGG	Y	3221138	6498	0	0	0	0
KRKNGM	Y	12032732	27288	0	0	0	0
KRKNMM	Y	7654095	121216	0	0	0	0
KRKNG	Y	1762807	3275	0	0	0	0
KRKNM	Y	2605497	5200	0	0	0	0
KRKN	Y	380325	609	0	0	0	0
KRKCGG	N*	3322727	1379102	0	891	0	891
KRKCMM	Y	7913097	2969808	0	0	0	0
KRKCG	Y	1820350	2462	0	0	0	0
KRKCM	Y	2694400	91748	0	0	0	0
KRKC	Y	393327	7479	0	0	0	0

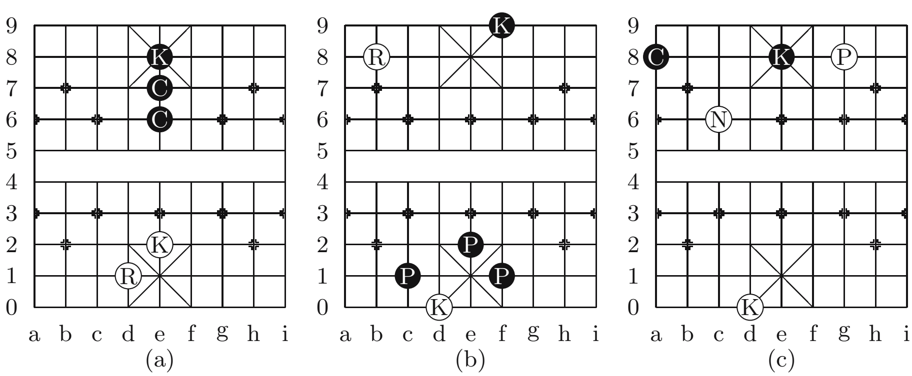

Fig. 3. Positions (a)(b) stained only by the Chinese rules and (c) stained by both rule sets

ing moves indefinitely, because Black has only one Cannon and the King. Note that Kings are allowed to chase. However, Figure 3(c) illustrates a surprising example. The game continues cyclically with the moves Nc6-b8 Ke8-e7 Nb8-c6 Ke7-e8, etc. Black loses the game because he chases a red Pawn indefinitely.