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, 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.
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, $DB: V \rightarrow \{\text{win}, \text{draw}, \text{loss}\}$. Each non-terminal position $u \in V$ satisfies the following constraints.

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

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 wins and losses, 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 draws.

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