An Overview and Classification of Retrograde Chess Problems

Marko Maliković
Faculty of Humanities and Social Sciences
University of Rijeka
Sveučilišna avenija 4, 51000 Rijeka, Croatia
marko.malikovic@ffri.hr

Abstract. *Retrograde chess analysis can be applied to several very different chess problems. These problems are often mutually so different that we can say that they belong to different domains. In the existing literature, there are no overviews or classifications of. As a result, under the names "retrograde chess problems" and "retrograde chess analysis" only small subsets of many types of problems are considered. In this paper we give an overview of retrograde chess analysis and our classification of retrograde chess problems. We also give an overview of various computer approaches for each of the retrograde chess types of problems.*

Keywords. Retrograde, Chess, Analysis, Problems, Classification

1 Introduction

In general, retrograde chess analysis is a method that determines which moves have been or could have been played leading up to the given chess position. Such methods can be applied to several very different types of chess problems. These problems range from the analysis of chess endgames to so-called "classical retrograde chess problems" that are designed especially for retrograde analysis. So, chess problems in which retrograde analysis can be applied are often mutually so different that we can say that they belong to different domains. By reviewing the existing literature, we can conclude that there are no overviews or classifications of. As a result, under the names "retrograde chess problems" and "retrograde chess analysis" only small subsets of many types of problems are considered.

In the last nearly fifty years there have appeared various computer approaches for each of the retrograde chess types of problems. Some of these approaches will be described in this paper. As we will see, for some types of problems the achieved results are satisfactory while for some types of problems the existing approaches are very doubtful and use specific, *ad hoc* methods and techniques. In any case, due to the complexity of the chess game, computer solutions for all types of problems have computational limits.

2 Classification of Retrograde Chess Problems

In this paper, we divide retrograde chess problems into two main groups: retrograde chess problems with practical applications in the chess game as such, and so-called classical retrograde chess problems as intriguing studies in pure deductive reasoning but without direct practical applicability to chess game playing.

I. *Retrograde chess problems with practical applications*. In this group, we include the following:

1. *Retrograde analysis of chess endgames* - If we apply the retrograde analysis in chess endgames with a limited number of pieces on the chessboard, then we can generate *tablebases* (database files of stored endgame positions), working backwards from the known outcomes (e.g. checkmate or stalemate). As we will see, these studies have directly influenced the rules of the chess game and the development of computer chess programs.
2. *Proving legality of the position* - The main task in these problems is to prove that a given position can be reached from the initial chess position in accordance with the rules of chess (as specified by the World Chess Federation - FIDE [34]). For proving legality of the position one can use *Proof games* and/or its special kind *Shortest proof games* [33].
3. *Castling and en passant problems* - Classical chess problems (e.g. Mate in two moves) are most often given without information about whether players can castle or can *en passant* capture the opponent's pawn. Through using retrograde chess analysis, it is sometimes

possible (considering the history of the position) to determine if castling is disallowed, and whether an *en passant* capture is possible.

II. *Classical retrograde chess problems*. These problems are essentially a matter of logical and combinatorial reasoning but have not any practical application in chess playing. The greatest value of such problems is usually the beauty of their queries and their solutions and development of (heuristic) methods for their solving which can later be applied on other types of problems. For illustration, here we list just some of these problems:

1. What were the last *n* moves?
2. The piece has fallen from the chessboard. From which square?
3. Which piece is represented by a coin lying on the chessboard (because players lost the original piece)?
4. Which piece on the chessboard is a promoted piece?
5. On which side of the chessboard is a white (black) player?

It is clear that these problems can take many other forms and that various questions about a given position can be asked. Among other sources that describe the classical retrograde chess problems, it is unavoidable to mention two wonderful books of Raymond Smullyan [21], [22]. These books show how beautiful and interesting retrograde chess problems can be, and have served many authors as the basis for developing methods for solving such problems.

On, perhaps the most well-known web site of retrograde chess analysis, "The Retrograde Analysis Corner" [14], there are some types of problems that we do not include in this paper. For example, *coloring problems* are problems in which one does not know whether the depicted pieces are black or white. One has to find out the colors, knowing that the position is legal. In this paper, we will not deal with them because such types of problems do not appear in real-world chess situations.

3 Retrograde Analysis of Chess Endgames

Endgame tablebases are computerized databases that contain precalculated exhaustive analysis of a chess endgame positions, involving a small number of pieces. The general method is to work backwards from mating positions or known winning positions. The practical application of this method in chess engines is that if a position documented by the database occurs in the game, then the engine can stop its search and reasoning process and can simply follow a move sequence from database. On the other hand, the theoretical significance lies in the fact that in this way can be discovered (formerly unknown) properties of some endgames.

Chess endgames were analyzed long before the era of computers. According to [24], human analysis appears from at least the ninth century with analysis of endgames ♔♖♔♘ and ♔♖♘♔♖.[1] The rules of chess were slightly different in those days (as stalemate was not necessarily considered a draw) and modern chess is generally considered to have begun roughly in fifteenth century.

In this paper we focus on computer analysis of chess endgames. In 1965, Richard Bellman was the first who proposed the creation of a computerized database to solve chess endgames, analyzing games backward from positions where one player is checkmated or stalemated, instead of analyzing forward from the current position [3]. That paper is theoretical (there is no associated computer implementation), although it describes possibilities of application of dynamic programming to chess. Even more, Bellman wrote: "Even with the techniques described above, we cannot handle king-piece-pawn endings with the computers currently available. It seems reasonable to predict, however, that these techniques will be powerful enough with the computers available within ten years or so." Bellman was a relatively good forecaster. The first practical steps in this direction were made in 1970 by Thomas Ströhlein published in his doctoral thesis [25]. Ströhlein developed a computer algorithm for generating all optimal games of several classes of endgames with three or four pieces. Several researchers have continued to work on the extension of the tablebases for the four and five piece endgames, including Ken Thompson and Lewis Stiller.

A good example from this period is Thompson's database for ♔♕♔♖ endgame from 1977 [17]. In general, this endgame is win for white, but it is very difficult for white if black plays optimally [15]. The longest winning sequences require 31 moves. Thompson's database was used against chess grandmaster Walter Browne, one of the best US chess players at the time. Brown was given a time limit of 2.5 hours to play up to 50 moves, in accordance with the FIDE's fifty-moves rule [34]. However, Brown was unable to succeed against the database in the required number of moves. After that, Brown carefully studied the computer's play. A few weeks later he played a rematch from another 31 moves position and this time he won, but exactly on 50 moves. Starting position in second game is shown in Fig. 1. Since then, many more grandmasters have

[1] In the main text of this paper we will use graphic symbols of chess pieces of an endgame. Also, we do not use symbols for black pieces, but only for white. For example, ♔♖♔♘ will represent the endgame of white king and white rook against black king and black knight (following the notation by Stiller [24]).

failed to win in winning positions, including world chess champions Garry Kasparov and Anatoly Karpov. These examples show great practical and theoretical significance of the computerized endgame tablebases.

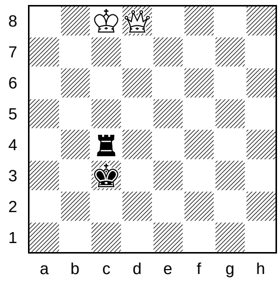

Figure 1. Position that requires 31 moves to white's win.

In 1986, Thompson published the first analysis of all endgames with five or fewer pieces [27], and in the late nineties, several six piece endgames were constructed [23], [28]. The benefits of these results were also numerous. The obtained tablebases resulted in overturning many human pre-conceived ideas. Some positions that humans have in the past considered as draws were proved winnable, but with the proviso that a tablebase analysis found a mate in more than a fifty moves which violate the fifty-moves rule. For example, ♔♕ beats ♔♗♗ (since 1634 believed to be draw) because tablebases had uncovered positions in this endgame requiring 71 moves to win. Another example is that ♔♗♗ versus ♔♘ is generally not a draw (as it was long time believed), and so on. This resulted in changes to the fifty-moves rule in chess. Actually, in accordance with the obtained results, FIDE changed the rule several times, to allow more moves for endgames where fifty moves were insufficient to win. For example, in 1988, FIDE allowed seventy five moves for ♔♗♗♔♘, ♔♘♘♔♙, ♔♕♔♗♗, ♔♕♔♘♘, ♔♖♗♔♖, and ♔♕♙♔♕ with the pawn on the seventh rank. After several changes, and following Stiller's discovery from 1991 that ♔♖♗♔♘♘ endgame has the maximum depth of 223 moves [23], in 1992 FIDE canceled exceptions and restored the fifty-moves rule to its original standing [12]. Thus a tablebase may identify a position as won or lost when it is in fact drawn by the fifty-moves rule. A complete and precise history of the fifty-moves rule can be found in [11].

Thompson's databases (along additional databases supplied by Stiller) were used in Deep Blue, chess machine that defeated then-reigning World chess champion Garry Kasparov in a six-game match in 1997 [5]. The endgame databases in Deep Blue included all chess positions with five or fewer pieces on the board, as well as selected positions with six pieces. All six piece endings were solved later in Nalimov tablebases [20], [4]. Nalimov tablebases are now used by many professional chess programs and services.

In 2013, complete tablebases which give optimal play for all endgames with seven or less pieces were generated by Zakharov and Makhnichev from the Lomonosov Moscow State University and are called "Lomonosov endgame tablebases" [6].[2] As it was expected, in these databases new interesting facts were found. Among others, the longest mating position for seven pieces is found. In the position showed in Fig. 2, black is to move, and white can mate in 545 moves.

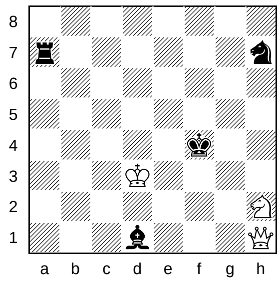

Figure 2. Longest mating position for seven pieces.

4 Proving legality of the position

According to some authors, retrograde chess analysis had been unknown to players around thousand years ago [33]. But, the precise year or even decade of the discovery of retrograde chess analysis is unknown. However, its occurrence can be naturally linked to the appearance of needs for proof of legality of positions given in classical chess problems (e.g. Mate in two moves) published in chess and other publications. The validity or invalidity of the given chess position is often possible to prove only with the help of retrograde chess analysis. In particular, this applies to chess positions published before the development of such analysis. It was noticed that some of published problems have illegal position, which is not allowed according to the Codex for chess composition [26]. A good example is shown in Fig. 3 [32].

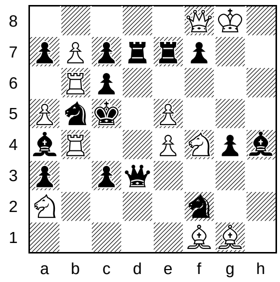

Figure 3. Mate in two moves.

[2] The size of this database is 140 Terabytes, which is obviously too much for personal computers. The database is accessible online.

White's pawn capture was *dxe* or *fxe*, explaining missing black piece. Black's pawn captures were *bxa*, *d7xc6*, *exd* and *dxc*, explaining all missing white pieces. But, there is no explanation for the disappearance of the white pawn from *g2*, so the position is illegal.

As already mentioned in Introduction, for proving legality of the position, *Proof games* and/or its special kind *Shortest proof games* can be used:

- *Proof games* - The problem of proof games is search for the any sequence of moves leading from the initial to the given chess position.
- *Shortest proof games* - Shortest sequence of moves leading from the initial to the given chess position. Although any proof game is a solution of the problem, searching for the shortest proof game (as a constraint) is often helpful.

Some authors, in order for the problem to be sound, require that proof games and shortest proof games must be unique [8]. However, if we focus on the practical aspect of these problems, this request is not critical.

There are several computer programs for solving (shortest) proof games but there is no scientific paper in this field. *Euclide* [7] and *Natch* [29] are the most well-known programs which are free to download and easy to run. Brief descriptions can be found on their websites. Euclide and Natch have fairly complex built-in algorithms and will not be described in this paper. However, in Fig. 4 we present a problem which can be solved by both programs [32].

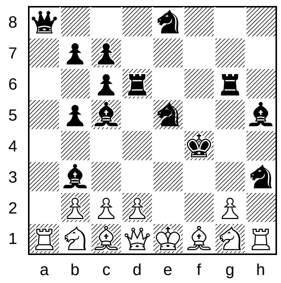

Figure 4. Position after the 32th move of white. How did the game go?

This problem is very difficult (because of multiple promotions and captured promoted pieces), but Euclide and Natch can find a solution. The solution is: *1. e4 a6 2. Bb5 axb5 3. h4 Ra6 4. h5 Rg6 5. h6 Nf6 6. hxg7 h5 7. a4 h4 8. a5 h3 9. a6 h2 10. a7 hxg1=N 11. Ra6 Nh3 12. Rc6 dxc6 13. e5 Kd7 14. e6+ Kd6 15. exf7 e5 16. f4 e4 17. f5 Ke5 18. g8=B Bc5 19. f8=N e3 20. Bc4 Be6 21. a8=R Nbd7 22. Ra1 Qa8 23. Nh7 Rd8 24. Bf1 Ne8 25. f6 e2 26. f7 exd1=B 27. f8=Q Bh5 28. Qf3 Bb3 29. Qd1 Kf4 30. Ng5 Ne5 31. Nf3 Rdd6 32. Ng1*.

This result does not mean that Euclide and Natch are able to solve every shortest proof game problem. There are many kinds of positions where they could not terminate in any reasonable time.

5 Castling and *en passant* problems

The game of chess is enriched by existence of moves whose legality depends on the history of the position. There are only two such moves: castling and *en passant* capture. The right to castle has been lost if the king and/or rook already moved, while *en passant* capture is permitted only if the last opponents' move was the double step of the pawn across the square which is attacked by players' pawn. But what in the case if position is given but its history is unknown? According to the abovementioned Codex for chess composition, rules that apply in that case are as follows:

(1) *Castling convention*: Castling is permitted unless it can be proved that it is not permissible.
(2) *En passant convention*: An *en passant* capture on the first move is permitted only if it can be proved that the last move was the double step of the pawn which is to be captured.

As already said in Section 2, classical chess problems (e.g. Mate in two moves) are most often given without information about whether players can castle or can *en passant* capture the opponent's pawn. A simple example of such problem, published in [14], is shown in Fig. 5.

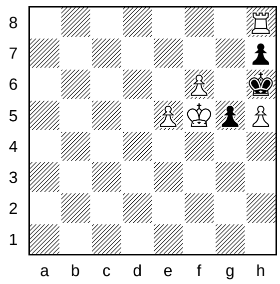

Figure 5. White to mate in two moves.

The last move of black could not be the king from *g7* to *h6* since the white pawn on *f6* could not have just moved to give check. Also, it could not be the pawn from *g6* to *g5* because in this case the white king was in check, and it was black's turn. So, black's last move can only have been pawn from *g7* to *g5*. Therefore, *en passant* capture is legal and after *1. h5xg6ep Kh5 2. Rxh7#* black is in mate.

However, the rules (1) and (2) from Codex for chess composition are not sufficient to solve all the problems related to the *en passant* capture and castling, as it is well elaborated by Keym [16]. In order to show that, let us consider the situation shown in Fig. 6.

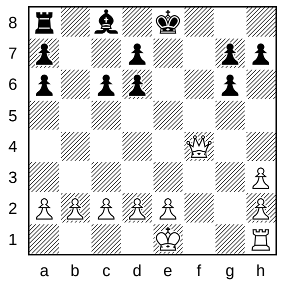

Figure 6. White to mate in three moves.

If we assume that white can castle, then the white queen on *f4* was promoted and retrograde analysis reveals that black cannot castle (the promoting white pawn from *f2* must have disturbed the black king). On the other hand, black can castle only if the white queen on *f4* was not promoted so if white cannot castle. This leads to problems because while neither white castling nor black castling can be shown not to be legal, white and black castling cannot both be legal. For these reasons, Codex was extended with the following two rules:

(3) *Partial Retrograde Analysis (PRA) convention*: Where the rights to castle and/or to capture *en passant* are mutually dependent, the solution consists of several mutually exclusive parts. All possible combinations of move rights, taking into account the castling convention and the *en passant* convention, form these mutually dependent parts. If in the case of mutual dependency of castling rights a solution is not possible according to the PRA convention, then the Retro-Strategy (RS) convention should be applied: whichever castling is executed first is deemed to be permissible.

(4) Other conventions should be expressly stipulated, for example if in the course of the solution an *en passant* capture has to be legalized by subsequent castling (a posteriori convention AP).

In the following example, published in [13] and shown in Fig., 7, application of the rule (3) is illustrated.

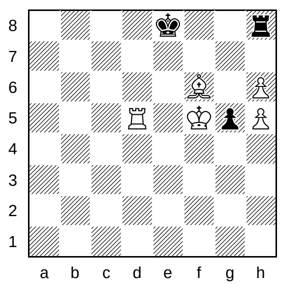

Figure 7. White to mate in two moves.

Either black can castle or black cannot. If black can, then black's last move was *g7-g5* and white can play *1. h5xg5ep* and then, after any black's move, either *2. Rd8* or *2. h7* leads to mate. If black cannot castle then *1. Ke6* leads to mate. So, it is impossible to determine what move black played last, and two options exist. According to the rule (3), both options have to be considering as exclusive parts. But it is interesting that the problem is well defined because both options lead to mate in two moves.

6 Classical Retrograde Chess Problems

Classical retrograde chess problems are more difficult semantic problems and for solving them there are often special heuristics developed. Therefore, the development of solving methods for such problems goes the other way than in problems described so far. For illustration, let us consider the Smullyan's problem [21], shown in Fig. 8. The question is whether white can castle.

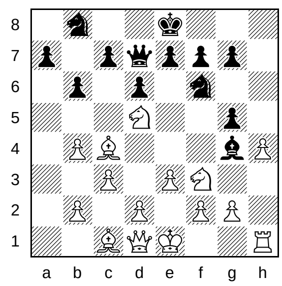

Figure 8. Can white castle?

White is missing only a rook and black is missing two rooks and a bishop, which was captured on its own square *f8*. Therefore, the pawn on *b4* captured the black rook and the pawn on *g5* captured the white rook. Black must have captured first, since prior to the capture neither of the black rooks could have got out on the board to be captured by the white pawn. How then did the missing white rook get out on the board to be captured by the black pawn prior to the white pawn on *b4* capturing? The only possible answer is that the rook on *h1* must really be the queen's rook! The sequence was this: first the king's rook got out and was captured by the black pawn, letting out the black rook to be captured by the white pawn. Then the rook from *a1* came round to *h1*. So the rook on *h1* is really from *a1*. Thus, white cannot castle.

In 1979, in one of the first papers that describe some computer solutions which deal with classical retrograde chess problems, Robert Filman made some attempts at formal representing of some knowledge required for solving one particular but very difficult problem [9]. Fig. 9 illustrates the problem that Filman's paper deals with. The problem is that the piece has fallen off of the chessboard from the square *h4* and the question is what piece was it. The position in Fig. 9 was achieved in a legal, not necessarily good-quality chess game.

To solve this problem, Filman created a first order logic formal deductive system in programming language LISP. This system was then extended to include some *observational* facilities. Namely, Filman stated [9] that generally intelligent systems needs to reason not only by *deduction* but also by other schemes such is *induction*, *analogy* and by *immediate recognition of results*, a process we identify as *observation*. For example, observations are inferences of the form "Black is in check" while deductions are inferences of the form "Both sides can't be in check at the same time, black is in check, therefore white is not in check". So, Filman concluded that problem showed in Fig. 9 requires both deductive and observational inferences.[3]

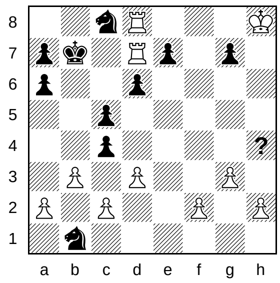

Figure 9. A piece has fallen off of the chessboard from the square *h4*. What piece was it?

It can be proved that the missing piece is the white bishop but the explanation of the solution to this problem is extremely complex and long [10], [21] and so out of the scope of this paper. Most of the facts that need to be observed in order to solve this problem are based on high level heuristics (similar to the problem showed in Fig. 8).

Filman concluded that solution of problem showed in Fig. 9 was beyond the ability of any computer program at that time. Anyway, in his first order logic system, he proved several lemmas about problem (using a proof checker for first order logic), which together provide a complete first order logic proof of the solution of the problem. Of course, this is an *ad hoc* proof, adjusted for a given problem, and cannot be applied to other problems. The complete proof can be found in his doctoral dissertation [10], in which consideration was given not only to the necessity for these particular choices (and possible alternatives) but also the implications of these results for designers of representational systems for other domains.

RETRO, an expert system for solving retrograde chess problems by means of heuristic methods was described in 1986 [1], [2]. Similarly as Filman, the authors focused on the difficult semantic problems such as Smullyan's problem [21], shown in Fig. 10.

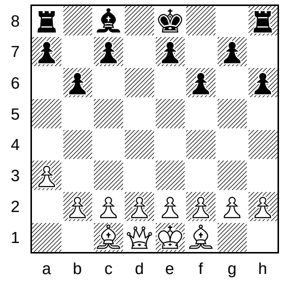

Figure 10. It is black's move. Can black castle?

Solution to the problem shown in Fig. 10 is much easier than in Fig. 9:

White's last move was clearly with the pawn. Black's last move must have been to capture the white piece which moved before that. This piece must have been a knight, since the rooks could not have got out on to the chessboard. Obviously none of the black pawns captured the knight, and black queen's rook could not have captured the knight, because there is no square that the knight could have moved from to get to that position. Likewise, the bishop couldn't have captured it, since the only square the knight could have come from is *d6*, where it would have been checking the king. Hence either the king or the king's rook has made the capture. So black can't castle.

The knowledge representation utilized by system RETRO is based on the concepts named *significant events*. Some of the significant events considered by RETRO are for instance: "King is in check" or "Pawn has promoted". Of course, a set of rules based on significant events in system RETRO is limited and have been derived from a consideration of the problems in Smullyan's book. So, as the authors themselves say, RETRO cannot solve any conceivable retrograde analysis problem. Authors also say that this approach is designed to be of general applicability, but it is quite likely that extra rules will be needed as more problems are considered.

In 1990, Karen White described the first PROLOG system developed for retrograde chess analysis [31]. Similarly as Filman's system, this system was capable of analyzing concrete types of retrograde chess problems using heuristics and modeling cognitive functions utilized by human problem solvers. The problems from [21] were used again in the design of the system but also cannot solve all problems from the book. Thus, this system is also limited to a specific subset of problems and it was planned to be upgraded with new heuristics for the application to new problems.

In addition to the abovementioned computer systems with *ad hoc* solutions to specific types of problems, we are aware only of one retrograde chess

[3] Furthermore, it was noted [9] that in some chess problems, semantically we have to go even further. For example, "the white queen's knight is not the same as the white king's knight", "a promoted pawn is not a queen", and so on. Generally, "values are different than the chess pieces themselves".

program of general purpose. *Retractor* [30] is developed in 1991 in the Department of Computer Science at Stanford University, California. Retractor uses a simple, classical backtracking search. All possible retrograde moves are generated at each node, with backtracking when a position is hit that can be proven to be either illegal, or previously reached. If the search reaches an implied given maximum depth without hitting a position it can prove illegal, then that branch is counted as a solution. But this does not guarantee that the solution is correct, only that the preprogrammed ruleset isn't able to prove the position illegal.

Also, there is a formal system for reasoning about retrograde chess problems using Coq - a formal proof management system [18], [19]. In this system a variety of heuristics to recognize some of the common chess patterns and to speed up solving problems were implemented. Due to these general heuristics, it can be applied to various types of retrograde chess problems. Given that the mentioned system is developed using a proof assistant, their advantage is that the results can be considered trusted. Of course, the additional advantage is that in Coq all the chess and heuristics rules are set on a declarative way, as the user does not have to provide a solving mechanism for the given problem. But its biggest drawback (also because it is developed using a proof assistant) is the slowness and limitation in search depth.

7 Acknowledgments

I thank Predrag Janičić for carefully reading this paper and providing valuable advices.

References

[1] Alden, B. E. P. *An expert system approach to retrograde-analysis*, Ph.D. Thesis. The Open University, 1986.

[2] Alden, B. E. P; Bramer, M. A. An expert system for solving retrograde-analysis chess problems. *International Journal of Man-Machine Studies*, 29(2):97-112, 1988.

[3] Bellman, R. E. On the application of dynamic programming to the determination of optimal play in chess and checkers. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 244-247, 1965.

[4] Bleicher, E. Nalimov Endgame Tablebases, http://www.k4it.de, downloaded: December 22th 2013.

[5] Campbell, M; Hoane, A. J; Hsu, F. Deep Blue. *Artificial Intelligence*, 134(1-2):57-83, 2002.

[6] Convekta Ltd. Lomonosov Endgame Tablebases, http://chessok.com/?page_id=27966, downloaded: December 22th 2013.

[7] Dupuis, É. Euclide, http://lestourtereaux.free.fr, downloaded: December 22th 2013.

[8] Elkies, N. D. New directions in enumerative chess problems. *The Electronic Journal of Combinatorics*, 11(2), 2005.

[9] Filman, R. E. The interaction of observation and inference in a formal representation system. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 269-274, Tokyo, Japan, 1979.

[10] Filman, R. E. *The interaction of observation and inference*, Ph.D. Thesis. Computer Science Department, Stanford University, 1979.

[11] Haworth, G. M. Strategies for Constrained Optimisation. *ICGA Journal*, 23(1):9-20, 2000.

[12] Herschberg, I. S; Van den Herik, H. J. Back to Fifty. *ICCA Journal*, 16(1):1-2, 1993.

[13] Hochberg, B. *Outrageous Chess Problems*. Sterling Publishing, New York, USA, 2005.

[14] Janko, O; de Heer, J. The Retrograde Analysis Corner, http://www.janko.at/Retros, downloaded: December 22th 2013.

[15] Jansen, P. *Using Knowledge about the Opponent in Game-Tree Search*, Ph.D. Thesis. Carnegie Mellon University, Pittsburgh, 1992.

[16] Keym, W. Castling and En-passant capture in the Codex 2009, http://www.janko.at/Retros, downloaded: December 22th 2013.

[17] Kopec, D. Advances in Man-Machine Play. In Marsland, T. A. and Schaeffer, J. (Editors). *Computers, Chess and Cognition*, 9-32. Springer, New York, 1990.

[18] Maliković, M. *Developing heuristics for solving retrograde chess problems*, Ph.D. Thesis. University of Zagreb, Croatia, 2008.

[19] Maliković, M; Čubrilo, M. What Were the Last Moves?. *International Review on Computers and Software*, 5(1):59-70, 2010.

[20] Nalimov, E. V; Haworth, G. M; Heinz E. A. Space-Efficient Indexing of Chess Endgame Tables. *ICGA Journal*, 23(3):148-162, 2000.

[21] Smullyan, R. M. *The Chess Mysteries of Sherlock Holmes: Fifty Tantalizing Problems of Chess Detection*. Hutchinson, London, 1980.

[22] Smullyan, R. M. *The Chess Mysteries of the Arabian Knights*. Alfred A. Knopf, Publisher, New York, 1981.

[23] Stiller, L. *Exploiting symmetry on parallel architectures*, Ph.D. Thesis. Johns Hopkins University, Baltimore, USA, 1995.

[24] Stiller, L. Multilinear Algebra and Chess Endgames. In Nowakowski, R. J. (Editor). *Games of No Chance*, 29:151-192. MSRI Publications, Berkeley, California, 1996.

[25] Ströhlein, T. *Investigations on Combinatorial Games*, Ph.D. Thesis. Technical University of Munich, 1970.

[26] The World Federation for Chess Composition - WFCC. Codex for Chess Composition, http://www.wfcc.ch/1999-2012/codex, downloaded: December 22th 2013.

[27] Thompson, K. Retrograde analysis of certain endgames. *ICCA Journal*, 9(3):131-139, 1986.

[28] Thompson, K. 6-piece Endgames. *ICCA Journal*, 19(4):215-226, 1996.

[29] Wassong, P. The Natch home page, http://natch.free.fr/Natch.html, downloaded: December 22th 2013.

[30] Whipkey, C; Hwa, T. Retractor, http://xenon.stanford.edu/~hwatheod/Retractor, downloaded: December 22th 2013.

[31] White, K. F. *Artificial intelligence and retrograde chess analysis: The design of a production system core rule base and interpreter*, M.S. Thesis. Department of Computer and Information Sciences, University of Florida, USA, 1990.

[32] Wilts, G. Chess Problem Database Server, http://pdb.dieschwalbe.de, downloaded: January 9th 2014.

[33] Wilts, G; Frolkin, A. *Shortest Proof Games*. Privately published in Karlsruhe, Germany, 1991.

[34] World Chess Federation - FIDE. Handbook, http://www.fide.com/fide/handbook.html, downloaded: December 22th 2013.