Losing Chess: 1. e3 wins for White

Mark Watkins
University of Sydney,
School of Mathematics and Statistics
watkins@maths.usyd.edu.au

1. INTRODUCTION

The purpose of this writing is to report on the recent proof that Losing Chess is a win for White (so that the game is weakly solved in the game-theoretic sense). Since the naming of this class of games has differing schools of terminology, we state for definiteness that captures are compulsory (but the choice of the player on turn when there are multiple captures), that the King has no special characteristics (and promoting to a King is allowed), and that castling is not legal. This is also sometimes called Antichess. See Pritchard's encyclopedia [P, §10.9] or the Wikipedia page for more.

Unless otherwise stated, we use "joint FICS/International Rules" for resolving stalemate (including having no pieces left). Under International rules, the stalemated side always wins. Under FICS (Free Internet Chess Server) Rules, when stalemate occurs the side with fewer pieces remaining wins (regardless of type), with equal numbers yielding a draw. Under "joint" rules, a stalemated side wins if having fewer pieces, and otherwise the game is drawn. In particular, any win under "joint" Rules is a win under both FICS Rules and International Rules.[1]

In particular, we have shown that 1. e3 wins against any Black defense. Twelve of the 20 Black responses have been known for approximately two decades to be lost, and Ben Nye solved a 13th in February 2003. To the best of our knowledge, this was the state of affairs when we started our work in late 2011. We give a fuller run-down of each Black response and its history in §5.

We cannot claim to have made any great scientific advances in our work. Indeed, the project was started mostly as a hobby, and we largely used proof number search [A, K] throughout. The availability of formidable hardware combined with strenuous perservance seem to be more relevant to our work (not to mention some luck). In §3, we describe our methods a bit more.

1.1 Historical sources

There exist a number of piecemeal Internet sites that have various information about Losing Chess. However, many of the pages of interest have not been touched in a decade or more.

1.1.1 Pages of Fabrice Liardet

Fabrice Liardet is one of best (human) Losing Chess players in the world. His French language site has a wealth of information about the game. It seems that these pages were last modified in 2005. The most directly relevant pages for our discussion are the opening pages on 1. e3.

1.1.2 Pages of Cătălin Frâncu

Cătălin Frâncu is the author of the Nilatac program. The last changes seem to be no later than 2006 (though see §5.5). He also provides a useful browseable opening book and (since 2012) 5-unit tablebases with Colibri.

1.1.3 Pages of Vladica Andrejić

This URL is currently working. The information seems to date from no later than 2004. While his Encyclopaedia of Suicide Chess Openings is not very complete, it does list some historical information unavailable elsewhere. For instance, it notes that ASCP (the program of Ben Nye) refuted the Andryushkov Defence (1. e3 c6 2. Bb5 cxb5 3. b4!) on Feb 3, 2003.

[1] There is also the *vinciperdi* ruleset, where stalemate with pieces remaining is a draw, so that winning entails losing all one's pieces. However, this game tends to take on a much different flavour, as Black can try to draw via a strategy of blocking enough pawns, for this greatly increases the difficulty of White to lose everything. One can note that almost always a stalemated side has at most pawns remaining.

1.1.4 Pages of Carl Lajeunesse

It seems that the `suicidechess.ca` website of Carl Lajeunesse has also unfortunately gone dormant. It had an immense opening book (around a billion nodes, with 15% expanded), though one must be aware (as with some of the above) that it uses FICS rules. The efficacy of this book is also unclear, as it only proved 1. e3 c6 on Dec 6, 2009. His website layout provided much of the inspiration for our interface described in §3.3 below.

1.2 Acknowledgements

In addition to the above webpages, the author would like to thank Gian-Carlo Pascutto and Lenny Taelman for useful comments in the earlier stages of the project. Moreover, Klaas Steenhuis has become very involved over the last 3 years, and contributed many concrete suggestions regarding improvements. He has also done some related work in proving that various White opening moves lose (see §6). Ben Nye contributed some historical information, and both he and Ronald de Man gave guidance concerning tablebases.

2. HISTORICAL MILESTONES OF OUR PROJECT

- 2011 latter half: interaction with Frâncu's Nilatac program and Lajeunesse's website.
- 2012 early part: implemented proof number search following Nilatac, adapting public domain IvanHoe code for move generation and tablebase generation.
- early 2012: 1. e3 c6 re-solved, and 1. e3 Nc6 also completed (both under International Rules).
- late August 2012: 1. e3 b5 solved using a 6-core machine with 8GB of RAM.
- Aug/Sep 2012: Frâncu announces that he solved 1. e3 Nh6 under FICS Rules when testing his new laptop over the last week.
- Sep/Oct 2012: transfer of 1. e3 Nh6 proof to International Rules, and 1. e3 g5 solved also.
- the next year: slow progress, though at one point 1. e3 b6 looked hopeful.
- Sep 2013, major rewrite started: to allow transpositions for known won positions; to make cluster usage standard (typically 128 cores); to ease reconstructing proofs after pn-search proves the root node to be a win; and to switch to joint FICS/International Rules.
- early Apr 2014: rewrite completed, and previous proofs transferred to joint Rules.
- Jul/Aug 2014: breakthrough in 1. e3 e6 line after switching to Qxf7, solution follows soon after, upon building some 6-piece tablebases.
- Feb 2015: 1. e3 c5 is solved, leaving only 1. e3 b6.
- May 2015: switch to 1. e3 b6 2. a4 instead of 2. Ba6.
- Aug 2015, two difficult lines in 1. e3 b6 2. a4 are solved, leaving 1. e3 b6 2. a4 e6 3. Ra3 Bxa3 4. Na3 Qh4.
- next 14 months: much effort on trying to get 5. h3 to work.
- Oct 2016: it is (finally) realized that 5. a5 solves this last line rather easily.

The main project webpage is `http://magma.maths.usyd.edu.au/~watkins/LOSING_CHESS`.

3. SEARCH METHODOLOGY

We briefly describe our search methodology. One description would be directly in terms of PN^2-search [BUH] but I find a different explication to be more useful. We consider proof-number searches as a type of an evaluation function. For a given position P we search a certain amount of nodes in pn-search.[2] This pn-search computes two values (proof/disproof numbers) for any position searched,[3] and we are most interested in positions near the root. An evaluation for each child of P is determined, for instance as its proof/disproof ratio (Frâncu's idea), and these are then stored in a higher-level tree, with P's info updated. The data below P's children are discarded, though we also recursively considered (in place of P) child nodes whose subtree's size was $\geq 70\%$ of its parent's.

[2] At the beginning of the project it was 10^7 nodes or 10 seconds, while at the end we used 10^8 nodes or 100 seconds, where the time-based cutoffs would typically only occur in endgames with many transpositions (requiring much backtracking to update proof/disproof numbers).

[3] We initialized proof/disproof numbers with the simple "mobility" weighting, being respectively set as 1 and the number of legal moves.

We then allow the upper-level tree to grow, extending leaf nodes via pn-search in some best-first manner, such as minimaxing the ratios/evaluations up to the root and expanding a critical leaf. However, to introduce some randomness into the selection process, we in essence took an idea from how opening books (in orthodox chess) randomise their choices. We walked down the upper-level tree from the root node, choosing a child randomly according to a weighting[4] from its minimaxed ratio (compare [ST]). This idea also makes parallelization easy – just run N independent instances of the pn-search simultaneously on various leaf nodes.

3.1 Transpositions

For transpositions, in both the pn-search and the upper-level tree we chose to identify nodes with the same position only when the reversible move count was zero. This had the advantage of dispelling any loops, though of course it is not very optimal. Only in the major rewrite (see §2) when trying to solve 1. e3 e6 did we implement transpositions to known wins, and then only in the upper-level tree. Frâncu has noted that his Nilatac solver (now quite old) was also quite lazy in its handling of transpositions, and this creates difficulties in his attempts to verify some of our later proofs.

3.2 Performance of automated methods

The above formed the basis of our automated search process. For the first year or so of operation, we typically ran our implementation on a 6-core machine with 8GB RAM (so that six upper-level nodes were being expanded at any given time), and with pn-searches of size 10^7 we produced about 1 million upper-level nodes in an overnight run. By the end of the project, we were typically running on (say) eight 16-core machines (each 128GB of RAM, of which about 80-90GB would be used), with pn-searches of size 10^8.

One disadvantage of the ratio-expansion (which is perhaps a worry with any heuristically-based best-first expansion scheme) is that one can sometimes wander into a "well" (local extremum) where almost all White moves have a great advantage, but none easily lead to wins. This is typical when White has a large material advantage, and Black a lone King, but White needs to push a pawn (or two) to promotion before the final wipeout. We shall comment more on this phenomenon when discussing individual lines. Often the final proof size (in nodes) is not a good metric as to the underlying difficulty, as the question is whether there are rapidly winning White moves that quickly distinguish themselves in terms of the ratio heuristic.

Another quirk is that there is a rather notable tempo-advantage in ratio-expansion. Often an upper-level node with unexpanded children will have its ratio go up/down by a factor 5-10 (or more) upon expanding them. I was not able to come up with any easy solution to this, and its interaction with the 70% child-inclusion from pn-nodes (which would tend to alternate who was on move, particularly upon a forced capture) was another difficult aspect.

After some initial experimentation, we found it quite advantageous to declare a draw to be a win for Black. This had the advantage of clearing up a lot of repetition draws from the upper-level trees. Adding knowledge of the opposite Bishop draw[5] was also useful, and of course tablebases (described more below) were quite powerful. However, we would still occasionally run into the "well" problem of above, and indeed the final solving of an upper-level tree would often take much longer than might be expected from a naïve extrapolation.

3.2.1 Some possible improvements

One idea that we never got around to implementing (at either the upper level or the pn-level) was a killer heuristic at sibling nodes; for instance, one could increase the priority of a refutation move by adjusting the ratios. Also, pushing pawns is always a useful way to try to break out of shuffling, and additionally might be considered to be of higher priority. We also toyed with different power-laws (or varying the power based upon subtree size) in the ratio-based upper-level randomized tree walk.

Another idea (compare enhanced transposition cutoffs [ETC]): upon creation of an upper-level node, look at its possible children, and see if any (via transposition) are already known to give a result that proves the node. This should be easy to implement, but again I never found the motivation to do so. A similar task could be to avoid dominated lines. A final consideration is that the predictive value of proof/disproof ratio seems related to game phase, that is, a ratio of 100.0 with many pieces left on the board will often be solvable rather quickly, but the same ratio in a situation where Black has only a King and pawns is rather likely to just be a slow endgame win.

[4]Typically as the square of it, so that a child of ratio 25.0 would be 4 times more/less likely to be followed as one with ratio 50.0.
[5]This is where (say) Black has nothing but a Bishop left, and among White's remaining pieces is a Bishop of the opposite colour.

3.3 Human input

There were two major ways that the above search procedure was augmented by human input. The first was in the choice of which upper-level tree (stored in different files) would be the next to be considered. In some of the more difficult lines, we would have upper-level trees corresponding to sub-sub-sub-sub-variations, which tended to make the project rather onerous from the standpoint of data management.

The second enhancement was via a Java-based interface that allowed the user (namely myself) to choose what upper-level node in a given tree to expand next. This displayed the proof/disproof numbers and minimaxed ratio (actually the natural logarithm of the latter), and allowed one to queue positions to be searched (overriding the automation). As noted above, this was inspired by the `suicidechess.ca` website of Carl Lajeunesse.

As a rough estimate, the solving of 1. e3 b5 took about 2-3 core-years, about 2-3 work-months using the Java-based interface (some of which was rather mindless clicking, but most of it seemed motivated), and about the same amount of human time in code development, including learning enough about Losing Chess and previous programs so as to have an idea of how to proceed. The Java code itself was adapted from the "ComradesGUI", written by the developers of IPPOLIT (see below).

3.4 Tablebases

The efficacy of having at least some tablebases can be seen from the position with (say) a White King on d8 and a Black pawn on d5. This is a draw (Black will King the pawn), though the proof/disproof ratio from a pn-search of 10^7 nodes can be around 300 or so, as White has much more mobility (hence many more options) than Black, at least at first.

We thus decided to develop a program to generate tablebases for Losing Chess. This, of course, is not novel, though I could not find anything particular to International Rules that had been done. At first, we decided to adapt the RobboBases of IPPOLIT developer "Roberto Pescatore" (this seems to be a pseudonym). However, in the end we ended up being unable to use almost all the clever ideas it contained: for instance, the index-differencing was found to be too dependent on the king-structure of normal chess, so we chose to re-compute the index from scratch. Similarly, with the king-slicing unavailable, the SMP machinery was then seen as too unwieldy, especially as we had no plans to build 6-unit TBs (though see below). The concept of a BlockedPawn counting as one unit was also dumped, even though it should be even more valuable in Losing Chess (where pawns on adjacent files can also be counted in such a way, with a bit more work).

In the end, our code built the 4-unit TBs in around 2 hours, and the 5-unit TBs in a couple of weeks. A verification unit detected a few errors (with *en passant*, unsurprisingly), but these were then fixed.[6] After some consultation with Ronald de Man, we later built a selection of 6-unit TBs, again firstly with the proof of 1. e3 e6. Here we restricted to 5-vs-K where the side with 5 units had no Queens or Bishops (thus pawn promotions were always to one of KNR). Later we allowed a Bishop (but only one), and we also built the 4-vs-KK and 4-vs-KN tablebases. These are by far the most useful in solving, as White has many options, all of which look good to the mobility-based pn-searcher. In contrast, most 3-vs-3 positions tend to resolve themselves rather quickly by pn-search, as both sides must play close(r) to a narrow line to avoid quick defeat.

Following the RobboBases, we decided to use distance-to-conversion (DTC) as the metric. In the pn-search, only the 4-unit TBs were accessed, and these were read from a flattened array of 2 bits per entry (WDL or broken). This takes about 800MB of memory, and allows fairly fast access.

The larger TBs could presumably be accessed in the pn-search, at least near the root, via a compression scheme such as that described in [TB]. For normal chess, similar compression reduces the size of the 5-unit TBs to about 450MB (both the Shredderbases and the RobboTripleBases are about this size, so too the more recent Syzygybases of Ronald de Man), at the cost of some additional computational overhead in capture resolution. It is unclear to me[7] what the comparative size for Losing Chess would be; firstly, there are more endgames (as the King is no longer royal), and secondly the compression efficiency from capture resolution would likely differ (due to captures being mandatory). As one goal of our research was to provide final proof trees which needed only the 4-unit TBs, we did not pursue this avenue too deeply.

---

[6] As above, I do not know of any other source for Losing Chess TBs for International Rules. However, since we are proving wins rather than draws, an alternative method of verification is simply to expand all relevant tablebase positions until a terminal position is reached.

[7] The expert would be Ronald de Man, who has built complete 6-unit TBs under FICS rules and the DTZ metric (distance-to-zeroing of the reversible move counter, so pawn pushes reset it), and I think he has underlying WDL data in compressed form too.

3.4.1 Some 5-unit tablebase facts

The longest 5-unit tablebase loss[8] under DTC occurs in KPkrp, at 78 moves (also found by Nye in 2002). Others in the 2-vs-3 genre with a loss longer than 50 are KNkkr and KNkkn (both 54) and KRkkn and KNkrn (both 55). In the 1-vs-4 genre, there is Kkbnp at 67, Kbnnp at 56, and seven others with a win taking more than 50 moves. In the Appendix, we give some maximal 5-unit positions, with a mainline for each.

These can be compared to the longest 4-unit conversions, where Kkbn already has a loss in 71 (wKd1 bKa3 bBf6 bNa5), with Kkkn at 50, and 6 others above 40 moves. The longest with a pawn is 31 moves (Kknp),[9] while almost all the long 1-vs-4 losses contain a pawn for the winning side (Kkkbb at 47 moves is the longest that does not). It does not seem that adding a fifth piece increases the complexity (maximal depth) of conversion that much, at least compared to normal chess where the increase from 5-unit to 6-unit to 7-unit nearly doubles each step.

We also give some examples of full-point zugzwangs in the Appendix, again comparing to the 4-unit case.

3.4.2 Some 6-unit tablebase facts

As noted above, we also built a subset of 6-unit TBs, initially 5-vs-K where the side with 5 units had no Queens or Bishops (thus pawn promotions were always to one of KNR). Later we allowed a Bishop, and also built the 4-vs-KK and 4-vs-KN tablebases.

The longest 6-unit DTC in our selection is a loss in 93 (wKa1 bKc8 bBd2 bNc6 bPb7 bPa6), and the longest pawnless example is a loss in 86 (bKc1 bBd6 bNb1 bNa1 wKh6 wKe1).

3.5 Post-processing the finished upper-level trees

The desired result of the above process would be an upper-level tree that was completely solved. One then still needs, however, to expand this into a full proof tree. During the early years of our project, at this stage we simply re-traversed the upper-level tree (after identifying all identical positions, irrespective of the reversible move count), and then re-ran the pn-search solver on each terminal node,[10] copying over the proofs. By the end of the project, every time a pn-search solved a node as a win for White it would write the solution to disk, so that the solver no longer needed to be re-run in this post-processing stage.

3.5.1 Data structures

Our upper-level trees were stored in a rather bulky data format, using about 40 bytes per node (again this varied over the project lifespan; e.g. 64-bit hash fields were used only after implementing transpositions to known wins). Each node had proof/disproof numbers, fields for parent, child, and sibling nodes, and a somewhat hackish implementation of transpositions that used two fields. These were combined with a field for minimaxed ratio (which was not saved to disk, but computed on loading), a field to estimate the subtree size (not always used at the upper level, but occasionally useful when deciding which White move to retain when multiple wins were known) and a half-field (16 bits) for the move played to reach this node. Another half-field was reserved for extraordinary usage (marking a node as being in TBs, or won/lost without any search-backing). This allowed easy updating of proof/disproof numbers, sorting of children by ratio, and management of transpositions.[11]

Comparatively, the final proof trees use 6 bytes per node in a much more compressed format. For instance, if node N has children, the first child must be node N+1 (with then node N+1 pointing to any sibling, and so on). A similar method was used with next-siblings when a position transposes, and the child and transposition flags are themselves 1-bit fields above a 30-bit[12] node-number indicator (the other 2 bytes are to record the move that corresponds to the incoming arc). This does not allow easy manipulation of trees, and tasks such as backtracking from a given node to the root require some extra work. The main advantage of this format is that it is more condensed than the fuller format.

[8] This is a remnant of how the RobboBase code works – it saves whether a position is won, drawn, or lost-in-X. It also does not distinguish whether White or Black makes the conversion.

[9] Though Rknp is over 40 under International Rules, its maximal loss is only 17 under joint Rules.

[10] In a perfect world, this suffices to re-solve the upper-level node – however, due to various gremlins (for instance, pn-search takes as input a specific path to a position, and so its internal expansion procedure might slightly differ upon transposition), this was not always the case.

[11] We used the same data structure in the pn-search, which meant that our original standard pn-search of 10^7 nodes used about 400MB. Multiplying this by a factor of 6 for our trivial parallelisation, and adding the 800MB for 4-unit TBs, our typical overnight job would fit comfortably in 4GB. Similarly, running 16 pn-searches of size 10^8 takes about 64GB, with an additional overhead of (say) 16x1GB to store transpositions (after irreversible moves) in pn-search.

[12] This turns out to be just sufficient to contain our final proof.

3.6 Final tree verification

An important component in our suite of programs has been the verifier. This checks a number of tree properties, such as whether all Black moves are considered, whether hash-identified nodes are the same,[13] that terminal nodes are wins (White has no moves and wins in FICS, or the final position has 4 units and is a White win), and more. This found a number of problems at various stages of our work. The major disadvantage from an independence standpoint is that it uses the same move generation and tablebases as the main program.[14] Frâncu and Lajeunesse have been able to transfer some of our proofs into ones for FICS rules with their own searcher engines, giving another partial verification of our work.

The LosingGUI described in §3.3 was also adapted into a WinningGUI, that allows one to walk through a proof tree, also giving counts on the size of subtrees. There is also currently a browseable version of the proof here.

3.7 Accounting of nodes

The extent of our proof tree is to allow White to reach a "won" position, which for us either is one where White has no moves and also wins under FICS rules, or is a known won position (for White) in 4-unit TBs.

In the node counts we give below, we counted each unique position only once.[15] The alternative method would be to count each path-expansion, no matter how many times the underlying position occurs in the tree, and no matter whether we identified such nodes in the tree.[16] One reason that this latter method might be preferred could be that the arcs of the graph are labelled by moves, and these are typically stored (in the data structure) on the target node.

To give an idea of how this affects our counts, the final proof tree (currently) has 863301867 nodes and 66187848 transposition pointers. Of the nodes themselves, 804549638 are internal nodes with 58752229 terminal, and the latter subdividing into 23448460 positions in 4-unit tablebases,[17] 18117361 where White has no pieces left, and 17186408 where White is stalemated without losing everything.

The large percentage of internal nodes is typical of Losing Chess, as often a win will contain a long sequence of forced Black moves, which when alternated with solitary White moves, creates long chains of single-child nodes. In fact, 359417248 of the 413452861 internal black-to-move nodes (87%) are positions with only one legal move.

4. GENERAL COMMENTS ABOUT LOSING CHESS STRATEGY

In order to help the reader understand some of the tactics and strategies that appear in the positions given in the next section, we herein recall some basic Losing Chess principles.

Firstly, as with many games, having more mobility (i.e. choices) than the opponent is typically a good thing. For this reason, a common tactic in Losing Chess is to (carefully) reduce the opponent to one or two units (a pawn or a king), and then tightly control the situation to be able to give up all your remaining pieces in one fell swoop.

Bishops are thought to be the worst pieces, as they can often be forced to take a large army one-by-one. Rooks can be similar, though they are also the piece of choice (e.g. when promoting) for someone who is trying to win an endgame. Queens on the other hand can more often get out of forced capture sequences by simultaneously threatening multiple enemy units. Kings have some stability via their lack of long-range danger, and are usually the piece of choice for someone trying to draw (typically the side with less material). Knights can also occasionally be prone to being forced to make a number of successive captures, while their short range is often not as useful compared to their slowness and lack of tempo capacity. Pawns are typically pushed forward to gain space.

Some openings are typified by "Queen capture races", where each side tries to capture a lot of enemy pieces with the Queen, coming as close as possible to wiping out the opponent before losing the Queen. Usually if one side emerges from this type of sequence a piece ahead (say) and there are no other complicating factors, then the game will be won in the long run.

---

[13] We can also vary the underlying hash computation to make the chance of a collision be vanishingly small, or indeed zero.

[14] Much of our move generation code followed that of IvanHoe (again from the IPPOLIT developers), adapted suitably for Losing Chess.

[15] For the size parameter in pn-trees, with the above-mentioned 70% cutoff for copying data to the upper level, we did not bother with counting transpositions – thus if both g3 and g4 led to a forced response of hxg3, whichever White move was played first would get the whole subtree attached to its size parameter. This was seen to be adequate for our purposes at the time.

[16] As our discussion of transpositions might indicate, the proof "tree" is really a graph, though the former word still seems to be used.

[17] Our proof tree has 79840511 positions with 5 units on the board, and 90352974 with 6 units.

5. BLACK RESPONSES IN MORE DETAIL

In Table 1, we give the size of the proof trees we obtained (see §3.7 for more about accounting). However, there might be significant reductions still available.

b6	448616089	Nh6	17495191	a6	243154	Nf6	22888
c5	217046510	Nc6	10755950	f5	90297	g6	4489
b5	82086575	c6	2135917	h5	69132	Na6	3309
g5	45550426	f6	510661	h6	43271	d6	33
e6	43471723	a5	353391	e5	43276	d5	33

Table 1: Node counts in our proofs after 1. e3

5.1 The well-known

Black has 20 responses to 1. e3, with 12 of these being fairly easy to refute. Two of them, namely d5 and d6, are particularly trivial. All of these are folklore, having been known for some time. Our f6 solution became much larger when switching to joint Rules (the automated solver explored 2. Qh5 rather than 2. Ba6); however, such variances are not uncommon in independent solvings, and subsequently Steenhuis reduced it to 269636 nodes.

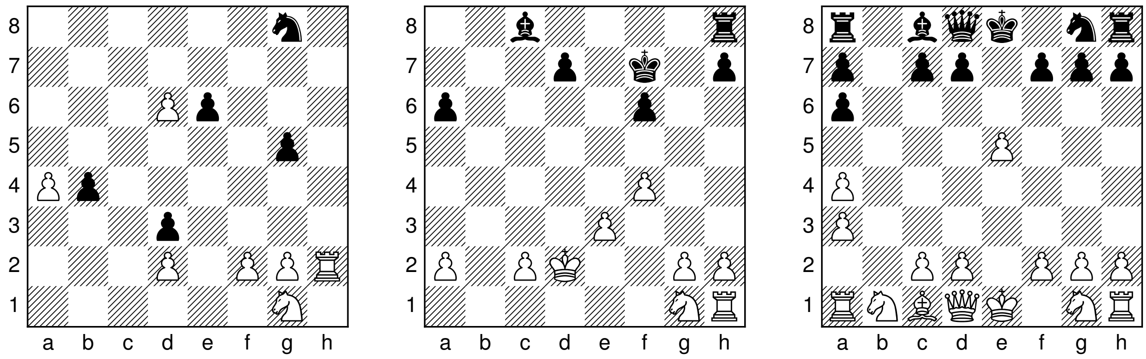


Figure 1: Lines from 1. e3 c6 and 1. e3 Nc6

5.2 1. e3 c6

As noted above, this seems to have been first solved by Ben Nye's program ASCP in February 2003, and a solution also exists in Nilatac's opening book. We largely copied over Nilatac's tree manually, but also found a simplification in the position given in Figure 1 (left). This arises after 1. e3 c6 2. Bb5 cxb5 3. b4 b6 4. Ke2 a5 5. bxa5 bxa5 6. c4 bxc4 7. Kd3 cxd3 8. a4 Na6 9. e4 Qc7 10. e5 Qxc1 11. Qxc1 Rb8 12. Qxc8 Rxb1 13. Rxb1 Nb4 14. Qxe8 g6 15. Qxf7 e6 16. Qxd7 Bd6 17. Qxh7 Rxh7 18. exd6 Rxh2 19. Rxh2 g5 20. Rxb4 axb4 (Figure 1 left). After 21. g4, Black has either b3 or e5 to put up any real resistance. We found that 21. g4 e5 22. a5! wins much more easily than Nilatac's 22. Rh7. Meanwhile, the try 21. g4 b3 22. Rh7 e5 23. Rb7 e4 24. Rxb3 Nf6 25. Rxd3 exd3 26. a5 Nxg4 27. Nh3 Nxf2 28. Nxf2 g4 29. Nxd3 g3 succumbs rather easily after 30. a6 (rather than 30. Nb2). One aspect is that Frâncu solved this in conjunction with Nye's 5-piece tablebases, which could yield quite bulky trees when expanded. Our original proof was under International Rules, and Klaas Steenhuis was able to verify (using Lajeunesse's website) that our simplifications largely transferred to FICS rules.

The above is the "mainline" (always choose largest subtree on Black moves) all the way to move 21 after 3... b6, though this itself is only 36% of the c6 solution, compared to 46% for 3... Qa5. For the latter, an exemplary line is 4. bxa5 a6 5. Qh5 b4 6. Qxf7 Kxf7 7. Ba3 bxa3 8. Nxa3 b6 9. axb6 Ra7 10. bxa7 Nf6 11. axb8=B e5 12. Bxe5 Bxa3 13. Bxf6 gxf6 14. Rc1 Bxc1 15. f4 Bxd2 16. Kxd2 (Figure 1 middle), with a subtree of size about 150000, and White will play Nf3, Rd1, f5, etc., with Black eventually running out of moves.

5.3 1. e3 Nc6 (Balkan Defence, according to Andrejić)

To the best of my knowledge, this was not previously proven to be a loss. However, it is not really *that* much more difficult than 1. e3 c6. The automated searching process yielded an abnormally large[18] proof/disproof ratio after about a core-week of running time, and then about 10-15 hours of manual work (some of it rather formulaic, such as "try all White moves here, and see if one wins") was sufficient to prove that White wins.

Our solution doesn't seem to follow any great patterns, as the main line 1. e3 Nc6 2. Ba6 bxa6 3. a4 Nd4 4. exd4 has about 81% of the nodes, and then 4... e5 has about 3 times as large a subtree as 4... Nh6. Black's first non-majority choice is at move 6 (after 5. dxe5 Ba3 6. bxa3, Figure 1 right), where Qh4 (29%), Nh6 (21%), a5 (16%), and Nf6 (13%) have significant subtrees.

[18] In retrospect, this was a bit misleading, as we had not yet implemented tablebases, and this omission tends to exaggerate said ratios.

5.4 1. e3 b5 (Classical Defence)

Chronologically, this was the next opening we turned to in our investigations, in part due to its great popularity in human play. Again the previous status of this opening is unclear to me. When I told Pascutto in 2012 that I was close to solving it, he seemed to remember a lecture about its resolution (or something related) some years ago.

Black here has three main tries, two of which then themselves split into three more main lines. We can note in passing that the "Suicide Defence", namely 1. e3 b5 2. Bxb5 Bb7, has long been known to be losing for Black.

5.4.1 1. e3 b5 2. Bxb5 Nh6

This defence is already about as complicated as 1. e3 Nc6 with our proof subtree having 8090747 nodes. The most difficult Black response is Nxd7 (45%), with Kxd7 and Bxd7 both around 27%. The mainline is now 1. e3 b5 2. Bxb5 Nh6 3. Bxd7 Nxd7 4. e4, whereas previously our proof used 4. c4 and was considerably larger.

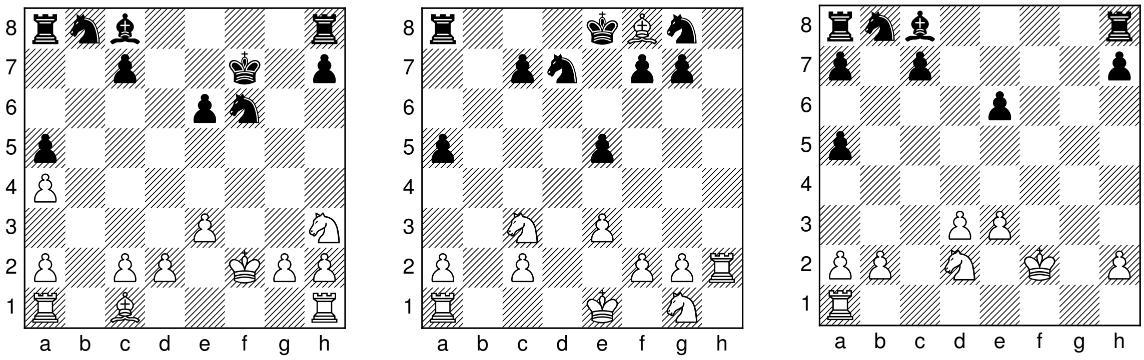


Figure 2: Lines from 1. e3 b5 and 1. e3 Nh6

5.4.2 1. e3 b5 2. Bxb5 e6

This proof subtree has around 24 million nodes. After 1. e3 b5 2. Bxb5 e6 3. Bxd7 Bxd7 4. Na3 Bxa3 5. bxa3, Black can complicate matters with Qh4 (59%), c6 (26%), or Bc8 (15%).[19] The former leads to 6. Qg4 Qxf2 (76%) 7. Kxf2 Bc8 8. Qxg7 a5 (57%) 9. Qxf7 Kxf7 10. Nh3 Nf6 (70%) 11. a4 (Figure 2 left) when Black has either Ra6 (49%) or Ke7 (29%) that have 1 million nodes or more in their subtree. White has some difficulty (particularly under FICS rules) with the doubled a-pawns, but in general has good mobility with c4, Rd1, e4, etc.

5.4.3 1. e3 b5 2. Bxb5 Ba6

This is Black's most lasting defence. After 1. e3 b5 2. Bxb5 Ba6 3. Bxd7 Nxd7 4. d3 Bxd3 5. Qxd3, each of Rb8, h6, and particularly Qb8 take substantial effort to defeat. The mainline of the latter, still having over 31 million nodes, is 6. Qxh7 Rxh7 7. Nc3 Qxb2 8. Bxb2 Rxh2 9. Rxh2 a5 10. Ba3 e5 11. Bxf8 (Figure 2 center), when either recapture leads to a subtree of around 15 million nodes. In either case, White plays Rh6, Black captures with the pawn, and then White plays f4 followed by a pawn exchange. This reduces it an endgame where both sides have a King, a Rook, two Knights, and four pawns. Perhaps a bit surprisingly, White is sufficiently better co-ordinated so as to win.

5.5 1. e3 Nh6 (Hippopotamus Defence)

Of the 7 unsolved lines from when this project began, Nilatac's book gave this one the highest proof/disproof ratio (that is, most likely to be a win). In fact, when we announced our results on 1. e3 b5 privately (on August 31, 2012), Cătălin Frâncu responded that he had recently shown (while testing a new laptop) that this line was indeed won for White under FICS rules, taking about two core-months of computing time.

His upper-level tree had 668 thousand nodes, which reduces to 167 thousand upon transposition-detection. We instrumented a utility to transfer his tree to our set-up. Upon simply attempting to solve all the upper-level nodes (including internal ones), this taking about 12 core-hours, we were left with only 15 unsolved nodes, upon which less than 10 minutes of manual work gave us a solved tree.[20] This was then expanded into a full proof tree as in §3.5, with the final node count around 17.5 million.[21]

[19] An unpublished guide to openings by Andrejić says Bc8 was the mainline before ASCP (of Ben Nye) proved it to be a loss, via 6. a4 Qxd2 7. Bxd2 (opposed to our 7. Kxd2). No dates are given.

[20] We tested our machinery by first re-solving 1. e3 c6 via similar importation of Nilatac's tree; in that case, our final proof tree had about 100000 nodes less than our first proof.

[21] This is the final size in joint Rules, though back in 2012 we originally transferred Frâncu's FICS proof to International Rules.

After 1. e3 Nh6 2. Ba6 bxa6 3. Qh5, Black has either g6 or c5, and c6 also lasts over 2.2 million nodes. In the first line, 3. Qh5 g6 4. Qxg6 fxg6 lasts 3 times as long as hxg6, with a thematic follow-up being 5. Ne2 Kf7 6. Na3 a5 7. g4 Nxg4 8. Rg1 Nxf2 9. Rxg6 Kxg6 10. Kxf2 Kg5 11. d3 Kf4 12. Nxf4 e6 13. Nxe6 dxe6 14. Nb1 Qxd3 15. cxd3 Bb4 16. Bd2 Bxd2 17. Nxd2 (Figure 2 right), and Black's queenside is too undeveloped to survive for long. The other mainline is 3. Qh5 c5 4. Qxh6 gxh6 5. b4 cxb4 6. Ba3 bxa3 7 .Nxa3 Qc7 8. Nh3 Qxc2 9. Nxc2 Bg7 10. Ke2 Bxa1 11. Rxa1 a5 12. a4 Na6 13. f3 h5 14. Ra3 h4 15. Kd1 f5 16. g3 hxg3 17. hxg3 Kf7 18. Ke1 Rf8 19. Kf2 Rh8 20. g4 fxg4 21. fxg4 Rd8 22. d3 (or Kg3) (Figure 3 left), and again Black's material advantage does not offset the lack of piece co-ordination (as can be seen by the Rook shuffles on the last few moves).

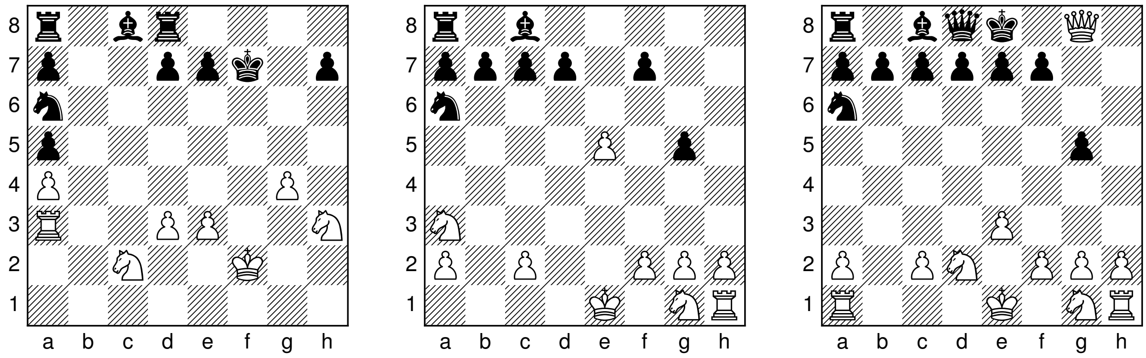


Figure 3: Lines from 1. e3 Nh6 and 1. e3 g5

5.6 1. e3 g5 (Wild Boar Attack)

With the above aid from Frâncu fortifying us that there might still be some relatively easy lines left to prove, we turned to 1. e3 g5, which had generally not seen much analysis. We first looked at 2. Bd3, but upon noting that 2. Ba6 bxa6 had been solved by Nilatac, we switched to this. Black's alternate try of 2. Ba6 Nxa6 took under a week to solve (about a core-month), with the final overall proof tree weighing in at 45.5 million nodes.

Almost 90% of the node-count is in the Nxa6 line, and there are two main variations after 3. Qh5 Bg7 4. Qxh7 Bxb2 5. Qxh8!, when both Bxa1 and Bxc1 have subtrees over 16 million nodes. In the first line, White plays 6. Qxg8, and then the Black's toughest defense is Kf8 (52%), with Bc3 also nearly 5 million nodes, and various other moves over a million. The mainline is then 6. Qxg8 Kf8 7. Qxf8 Qxf8 8. Bb2 Bxb2 9. d4 Bxd4 10. exd4 e5 11. dxe5 Qa3 12. Nxa3 (Figure 3 center), when b5 and c5 both take 3 million nodes to defeat, and Nc5 is also over 1.5 million. However, I suspect humans find the position already clearly winning for White, with just a long endgame to go.

If Black instead captures with 5... Bxc1, then the mainline is 6. Qxg8 Bxd2 7. Nxd2 (Figure 3 right), where b6, Nb8, and Kf8 are all over 4.5 million nodes, and c5 also above 2.4 million. White meets b6/Nb8 with f4 (after Qxe8), while Ne2 defeats Kf8, with again Black's two main moves being b6 and Nb8, the former met by f4 and the latter by Nb3 (in which case Black again prefers b6, with White playing Kd2, forgoing f4 in this sequence). White might also play 7. Qxe8 first and transpose in most variations.

The transfer of Nilatac's proof for 2. Ba6 bxa6 saw no problems, taking about 5000 of our upper-level nodes and 5.7 million nodes in the final proof tree. The mainline here is 3. Qh5 Bh6 4. Qxf7 Kxf7 5. e4 Qe8 6. e5 Bf8 7. Ne2 Nf6 8. exf6 exf6 9. a4 Qxe2 10. Kxe2, where both Ba3 and a5 still have around 500 thousand nodes left.

5.7 1. e3 e6 (Modern Defence)

As noted in our history (§2), there was a considerable rewrite of the underlying code, for multiple reasons including handling transpositions and allowing cluster usage. We increased the hardware by a factor of 20 or more since the solving of previous lines.

The mainline for some time (cf. Remmel-Liardet, Round 4 of the 2001 tournament [V39]) 1. e3 e6 2. b4 Bxb4 3. Qg4 Bxd2 4. Qxg7 Bxe3 5. Bxe3 c5 6. Bxc5 b6 7. Bxb6 Qxb6 8. Qxh7!? Rxh7 9. Nc3 Qxf2 10. Kxf2 Rxh2 11. Rxh2 Nh6 12. Rxh6 Ba6 13. Bxa6 Nxa6 14. Rxe6 fxe6 (Figure 4 left), with other possibilities in developing the Queen's Knight with 9. Nd2 or 9. Na3. Unfortunately, we were never able to complete a proof here even though it had a relatively high Nilatac proof/disproof ratio, as the endgame was quite slow-moving.

Nilatac suggested 8. Qxg8 was perhaps just as good, but in the end we found that 8. Qxf7! (or by transposition on move 7) was the best way to proceed. Again one finds that choosing the best moves in a heuristic sense (such as ratio) needs to be done carefully, and not overlook lines that might work out when searched deeper.

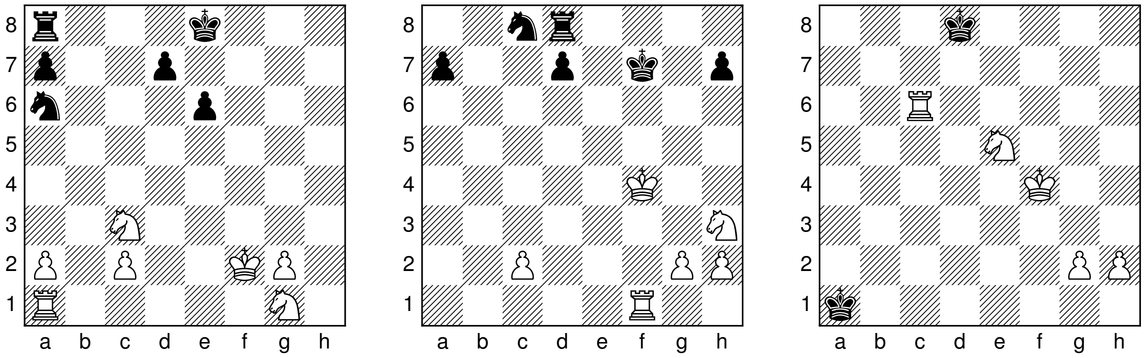


Figure 4: Lines from 1. e3 e6

White then proceeds with 9. Nh3 Qxf2 10. Kxf2 Ba6 11. Bxa6 Nxa6 12. a4 Nc7 13. a5 Ne7 14. Ke3, when Black has various choices (many of which transpose), but the most crucial is 14... Rhd8 15. Ra3 Nc8 16. a6 Nxa6 17. Rxa6 Rb8 18. Rxe6 Rxb1 19. Rxb1 Kxe6 20. Kf4 Kf7 21. Rf1 (Figure 4 center). The subtree here still has over 9 million nodes, and what occurs is that Black is soon reduced to a lone King, for instance after 21... Rf8 22. c4 a5 23. c5 a4 24. Nf2 a3 25. Rb1 Nb6 26. Rxb6 h6 27. Rxh6 Rh8 28. Rxh8 a2 29. Rc8 Ke7 30. c6 dxc6 31. Rxc6 Kd8 32. Nd3 a1=K 33. Ne5 (Figure 4 right), and the threat of Nd7 means that the Black Kd8 is lost. This is then a KRNPP vs K endgame, which can be solved by tablebases (though admittedly the path is not short, due to the g2/h2 pawns needing to promote in many instances). Indeed, we first had contacted Ronald de Man with an idea of building 7-unit TBs, but he indicated that his solver (which uses 6-unit TBs in pn-search) already could solve the position after White's 21st move, as there were no difficulties in reducing from 7 units to 6.

Our e6 proof is actually smaller than that for g5, but as indicated, was (significantly) more difficult to uncover.

5.8 1. e3 c5 (Polish Defence, or Goldovski Defence)

This was probably thought to be the hardest line, both by human impression and Nilatac's proof/disproof ratio. The automated prover actually did quite well with c4 (a subtree over 100 million nodes), g5, and Nh6, leaving Qc7 as the critical line. Here however, the heuristics did not work so well. Firstly after 1. e3 c5 2. Bb5 Qc7 3. Bxd7 Bxd7 4. Qf3 Qxh2 5. Rxh2 h6 it seemed that 6. Qxb7 was the way to go rather than Rxh6, but then the latter then started trending well after 6. Rxh6 Nxh6 7. Qxb7 Bh3 8. Nxh3 g5 9. Nxg5 Ng8 10. Nxf7 Kxf7 11. Qxe7 Nxe7 12. b4 cxb4 13. Ba3 bxa3 14. Nxa3 Bh6 15. Nc4 Bxe3 16. dxe3 Na6 17. f4 Ng8 18. Nb2 Rh6 (Figure 5 left). The main idea here was 19. Nd1 which looked completely winning (by Nilatac ratio) for quite some time, until a difficult drawing line was found: 19. Nd1 Nb8 20. Nf2 Rh1 21. Nxh1 Nd7 22. Ng3 a5 23. c4 Nf8 24. f5 Ne6 25. fxe6 Kxe6 26. Ke2 Kf5! 27. Nxf5 Ne7 28. Nxe7 Rc8 29. Nxc8 a4 30. Rb1 a3 31. Rb2 axb2 and draws (the pawns are not sufficiently advanced), or 24. Ne2 Nh7 25. Nd4 Ng5 26. fxg5 Ke6 27. Nxe6 Nh6!.

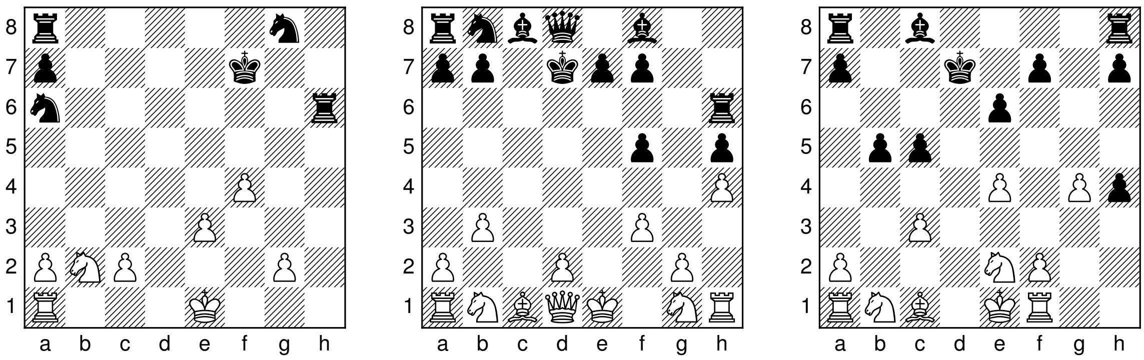


Figure 5: Lines from 1. e3 c5

Instead it turned out that 19. g3 did the trick, though this move was rather unimpressive in Nilatac ratio initially, and even with pn-searches of size 10^8 it took quite some time for real winning chances to become apparent. Indeed, we built 4-vs-KK tablebases to try to assure ourselves of the situation, and were then able to solve this line. It turns out that Qc7 is slightly smaller (17%) than Nh6 (20%). The position at move 19 contains 32.6 million nodes in its subtree, compared to 36.7 million in the entire Qc7 subtree. The mainline with 2... c4 is not so impressive, as after 3. Bxd7 Kxd7 4. b3 cxb3 5. cxb3 Nh6 6. Ne2 g6 7. e4 Nf5 8. exf5 gxf5 9. f3 h5 10. Ng1 Rh6 11. h4 (Figure 5 center), after any Black move the subtree proof size is under a million nodes. Comparatively, with 2... Nh6 3. Bxd7 Nxd7 4. e4 Ng4 5. Qxg4 g5 6. Qxd7 Kxd7 7. Ne2 Qb6 8. c3 Qxb2 9. Bxb2 e6 10. Rg1 b6

11. g4 b5 12. Bc1 Bg7 13. h4 gxh4 14. Rf1 Bxc3 15. dxc3 (Figure 5 right), Black still has h6, Kc6, and b4 all of which have proof subtrees larger than 2.5 million nodes.

It was not obvious to choose 2. Bb5 in the first place, and an extended testing process (making a upper-level tree with around 30 million nodes, with each underlying pn-search being of size 10^8) found this the most promising.

5.9 1. e3 b6 (Liardet Defense)

This left the final line 1. e3 b6. Already in 2013 I had thought this line might be solved, due to the try 2. Ba6 Nxa6 3. Qh5 c5 4. Qxh7 Rxh7 5. Kd1 Rxh2 6. Rxh2 e6 7. b4 cxb4 8. Rh6 gxh6 9. d3 b3 10. cxb3 Ba3 11. Bxa3 Nb4 12. Bxb4 a5 13. Bxa5 Rxa5 (Figure 6 left) when after a White move like 14. Kc2 we have 14... Rxa3 15. Rxa3 Ba6 16. Rxa6 Qg5! 17. Rxe6 Qxe3 18. fxe3 fxe6; however it turns out that Black reaches a draw by typical sequences like Kf7, h5, Ne7, Kg6, since White's movement is too slow with short-range pieces (KNP).

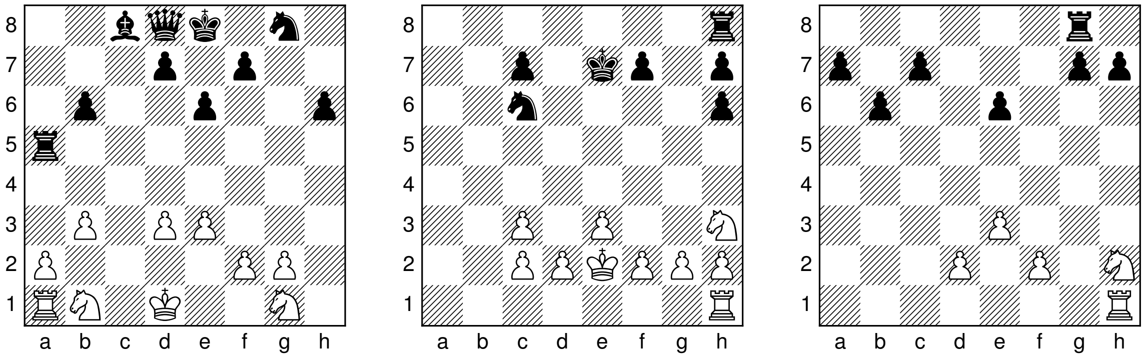


Figure 6: Lines from 1. e3 b6

There are other tries for White in the 2. Ba6 line (particularly at move 5), but after many failed attempts we switched to 2. a4 in May 2015. This was not exactly a move that had been considered too much in the Losing Chess community prior to this. Everything but e6 and b5 loses rather quickly (though Ba6 is 16.9 million nodes), and the mainline in the latter is 3. Bxb5 Nh6 4. Bxd7 Qxd7 5. Qh5 Qxa4 6. Rxa4 Bh3 7. Rxa7 Rxa7 8. Qxh6 gxh6 9. Nxh3, when the remaining subtree is 125 million nodes, with half of this involving Bg7, whose mainline goes 10. Nc3 Bxc3 11. bxc3 Ra3 12. Bxa3 Nc6 (50%) 13. Bxe7 Kxe7 14. Ke2 (Figure 6 center), and White's extra pawns eventually win.

This left the line 1. e3 b6 2. a4 e6 3. Ra3 Bxa3 4. Nxa3, where we found that Black's try 4... Qe7 loses to a capturing race after 5. b4 Qxb4 6. Qf3 Qxa3 7. Qxa8 Qxa4 8. Qxb8 Qxc2 9. Qxc8 Qxc1 10. Qxe8 Qxe1 11. Qxd7 Qxf1 12. Qxf7 Qxg2 13. Qxg8 Rxg8 14. Nf3 Qxh2 15. Nxh2 (Figure 6 right), where the resulting subtree is only 2.38 million nodes, though heuristically it is not immediately clear White should have good winning chances. A similar win came about in 4... Qh4 5. h3 Qxa4 6. Qf3 Qxa3 7. Qxa8 Qxb2 8. Qxb8 Qxc2 9. Qxc7 Qxd2 10. Qxc8 Qxe1 11. Qxe8 Qxe3 12. Qxf7 Qxh3 13. Rxh3 Nh6 14. Bxh6 gxh6 15. Qxh7 Rxh7 16. Rxh6 Rxh6 17. Bc4 b5 18. Bxe6 Rxe6 19. f4! which left only 5... Qxf2 in this line.[22] However, it would turn out that this was a false lead, as if we had similarly searched for a Queen capture race in 5. a5 bxa5 6. Qh5!, we would have achieved the final proof approximately 14 months sooner – instead, we wasted these months on various lines after 5. h3 Qxf2, making much "progress",[23] but also being entangled in slow-moving endings.

The sideline 4... Qf6 is not particularly difficult for a computer after 5. Bd3 Qxf2 6. Kxf2 b5 7. Nxb5 a5 8. Nxc7 e5 9. Bxh7 Rxh7 10. Nxa8 Rxh2 11. Rxh2 Kd8 12. Nc7 Kxc7 13. Rh3 (Figure 7 left) when there are 4.7 million nodes in the subtree. The principal line (by far) is 4... b5 5. Bxb5 Qg5 6. Bxd7 Qxg2 7. Bxe6 Bxe6 8. h3 Bxh3 9. Rxh3 (Figure 7 center) when Black can play either Qxf2 or Qxh3, the former nearly 150 million nodes and the latter just over 100 million. The mainline 9... Qxf2 10. Kxf2 h5 11. Rxh5 Rxh5 12. Qxh5 Ne7 13. Qxf7 Kxf7 14. d4 Ng8 15. e4 Nd7 16. Bh6 Nxh6 17. e5 Nxe5 18. dxe5 Rd8 19. c4 (Figure 7 right) still has a subtree of almost 55 million nodes. The other mainline is 9... Qxh3 10. Nxh3 Nd7 11. c4 Nf8 12. c5 Ke7 (39%) 13. f4 h6 (34%) 14. Qh5 Nh7 (also Rc8) 15. Qxh6 gxh6 16. f5 Ng5 17. Nxg5 hxg5 18. b3 Rf8 19. a5 Nf6 20. Nc4 Ng4/Nd5 21. Ke2 Nxe3 22. Kxe3 (Figure 8 left), when a 2.6 million node subtree remains and (e.g.) White beats Ke8 by b4 and Kd7 by a6, with eventually Nb6 being likely after the queenside pawns advance more.

[22] When creating a new directory for 1. e3 b6, we had erroneously linked to tablebases with only International Rules, and it was somewhat fortuitous we were still able to get various endgames to work out.

[23] The remaining unsolved line is 6. Kxf2 Na6 7. Bxa6 Bxa6 8. Qg4 Bc4 9. Qxg7 Kf8, though Black had many other ninth moves that we solved via significant effort, in fact in total size exceeding the rest of the proof!

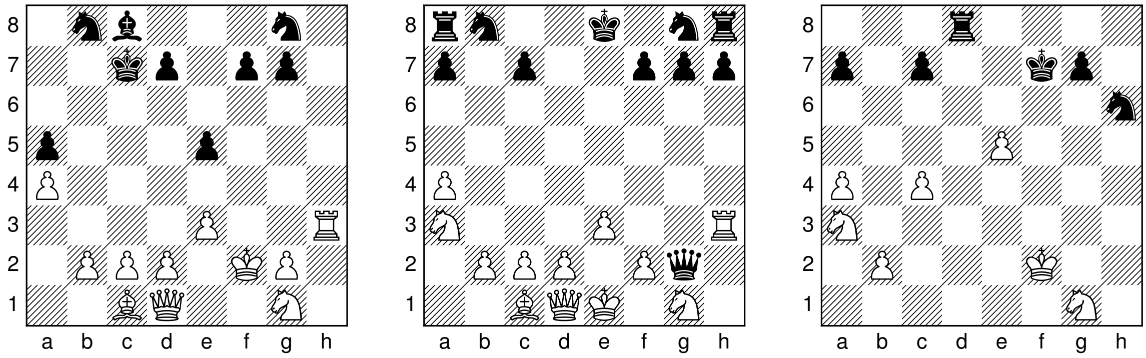


Figure 7: More lines from 1. e3 b6

This then left 4... Qh4 5. a5 to finally be uncovered, and once the idea of 5... bxa5 6. Qh5! was hit upon, the automatic solver took only a couple of hours to solve this (and also 5... Qxf2), finishing the proof that 1. e3 wins for White. The main obstacle was the line 6. Qh5 Qxf2 7. Qxa5 Qxe3 8. Qxc7 Qxe1 9. Qxc8 Qxf1 10. Qxe8 Qxc1 11. Qxf7 Qxd2 12. Qxg7 Qxg2 13. Qxg8 Qxg8 14. Nh3 Qg5!? 15. Nxg5 Re8 16. Nxe6 dxe6 (Figure 8 center) when one needs to expand some upper-level nodes before 17. c4 shows a promising proof/disproof ratio.[24]

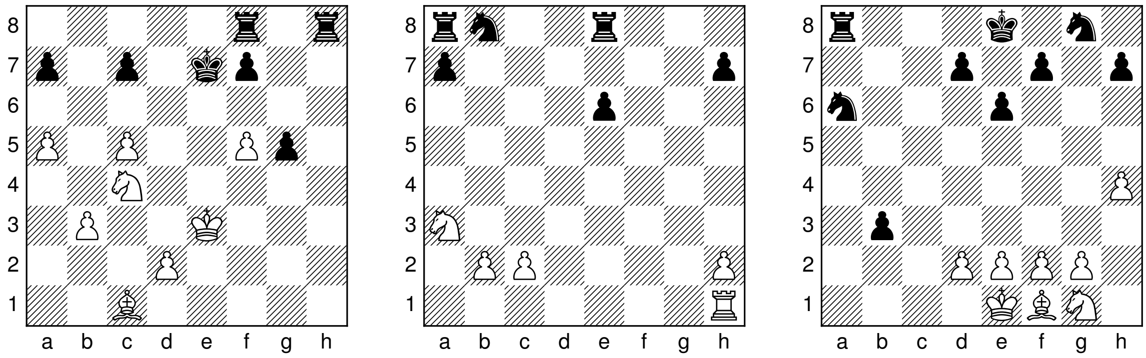


Figure 8: More lines from 1. e3 b6, and 1. a3 e6

6. OTHER INITIAL WHITE MOVES

I must admit to not knowing so much about this subject. Clearly 1. e4, 1. d3, and 1. d4 have been known to lose for decades. Nilatac also has fairly short proofs that Nc3, Nf3, f4, and h4 lose, and a longer proof (cited by Andrejić as due to Nye's ASCP in Jun 2003) that h3 loses.

Klaas Steenhuis has been quite interested in this subject, and over time prodded me to implement the ability to have "NULL" moves (and thus switch White and Black) in the solver software, so that he could explore more lines. Firstly he solved h3 more completely, extending Nilatac's opening book to a full proof. He then solved b4 as lost for White; this had been done by Nye in late 2002 [V41] soon after his implementation of 5-unit tablebases, but there was no proof easily available (Nilatac's book had none). Steenhuis then produced three other proofs, namely that c3, f3, and a3 all lose, the last by far the most complicated. Indeed, it has over 180 million nodes, making it nearly comparable to 1. e3 c5 in size.

The mainline is 1. a3 e6 2. c4 Bxa3 3. Nxa3 c5 4. b3 (60%) a5 5. b4 cxb4 6. h3 bxa3 7. Bxa3 Qf6 8. Be7 (69%) Qxa1 9. Qxa1 Nxe7 10. Qxg7 Rg8 11. Qxg8 Nxg8 12. h4 (68%) a4 13. Rh3 (71%) b5 14. cxb5 Ba6 15. bxa6 Nxa6 16. Rb3 (66%) axb3 (Figure 8 right), with a subtree of nearly 19 million. Black has a clear advantage, but converting this to a proven win is not very quick. White also has 4. d4 which has a subtree of almost 60 million, and Steenhuis notes this was (much) the more difficult line to solve in practice (Black gives up his King in the mainline, but still wins the endgame), and he also built some 6-piece tablebases to help the process.

7. FUTURE DIRECTIONS, AND FINAL COMMENTS

As we noted in the Introduction, we cannot be said to have approached the solving of Losing Chess in a particularly scientific manner, and indeed this was a cogent criticism of our previous article submitted to the ICGA Journal. However, now that that the proof is done, we possess a baseline for comparison. For instance, we can implement some of the ideas in §3.2.1, and compare how fast (or if at all) the automation techniques are successful. Any conclusion will be domain-specific to Losing Chess, but might still be of interest for general pn-search.

[24] In the end however, this subtree has under a million nodes, again showing that proof size and proof discovery can differ drastically.

One anecdotal observation we have made is that large pn-searches seem to work better than multiple smaller ones (for instance, one search of size 10^8 rather than ten of size 10^7 in the same time frame), and moreover this tends not to produce such large files for upper-level trees. However, this perhaps could be studied more, and in particular in how our usage of merely Nilatac ratio as a heuristic (opposed to enhancing it with other facets like game phase) affects the situation.

There is also the consideration of final proof size. Currently our proofs contain some rather unwieldly lines, particularly (due to our "50-move rule" implementation) when White shuffles around for 40-odd moves before pushing a pawn. These will likely be improved in the future. I certainly expect that 5% of the size of the final proof can be removed rather simply like this, and larger reductions coming about from alternate White moves may also be quite feasible.

Future explorations with randomized opening setups are also possible, and again Steenhuis has made experiments.

7.1 The complexity of Losing Chess

We are often asked how "complex" our work is, particularly with respect to the solution to checkers [S]. We find the games to be rather incomparable, as one is a draw, and the other is a win. Although one can find drawn games with rather short proofs,[25] they tend to be more bulky in final size. One reason for this is that drawn games are often "wandering" (long reversible sequences) or "loopy" (so analysis of repetitions becomes critical, cf. [GHI]), and thus the average path length until reaching a solved position (such as tablebases) can be much larger.[26]

Our final proof size is approximately 900 million positions, and even with much larger tablebases (full 6-piece) this would not be reduced by more than about 15%. Table 2 of [S] indicates that about 15 million searches of 15 (alpha-beta) or 100 (Df-pn) seconds each were done, with about 1/3 of these ending up being used in the final proof.[27] Comparatively, we searched for approximately 200-300 core-years at around 2 million pn-nodes per second, or approximately 10^{16} total positions searched (about 100 times as many as for checkers). While checkers has about $5 \cdot 10^{20}$ positions in its search space, presumably Losing Chess has many (many) more than this.[28] As a rough estimate, we did 10^8 pn-searches of size 10^8, and the end proof used maybe a million of these (1%), with approximately a 1:1000 expansion between upper-level trees and full proofs.[29]

7.1.1 Automating everything

The inevitable end of a scientific approach would be a complete automation of the solving process. As indicated above, this works fairly well for some of the easier proofs. However, we are of the opinion that there is still a considerable art (or lack of science) involved in the harder ones. It is impossible to estimate how much effort was saved by human intervention in our current approach, but the quick resolution of the final 1. e3 b6 line upon exploring a sideline "correctly" (meaning in human terms, "deeply enough to see what is going") indicates that our heuristics used in automation could be much improved (cf. the "wasted" effort in the last paragraph).[30]

7.2 Win versus Draw

One surprise to me in the aftermath of announcing the result is how many people (including chess grandmasters who had made some study of the game) told me that they had thought that 1. e3 would only draw, particularly against c5 or b6. On the other hand, my whole involvement in the project was a sort of "gamble" that White would win in the end, as else there would be little chance of completion.[31] I'm not sure when exactly I became convinced of this, but certainly after 1. e3 b5 was solved followed by Nh6 (Frâncu) and g5 in rapid succession I was rather optimistic, particularly since e6 and b6 both had decent scores in Nilatac's book. However, as progress could only be made by slow endgames in many of the resulting lines, the situation was not completely clear, especially as 2. Ba6 seemed not to work out against b6. Trying 2. a4 here was quite fortuitous, and at least by mid-2015 I was quite sure that White would win (though whether a proof of this would be feasible was not apparent). The final refuting of 4... Qh4 here was unfortunately quite delayed.

[25] An obvious example in orthodox chess would be when both players need to head to a repetition to avoid immediate loss.

[26] Checkers is somewhat an outlier since there are no reversible moves at the start, and by the time Kings appear the forced-capture rule has usually reduced the number of pieces so that tablebases are within sight (see also [S, §3.5]). In [S] the "upper-level" tree (in our terms) had a maximal depth of 94 ply, with possibly a 50 ply search then leading to tablebases which themselves could represent hundreds of ply.

[27] By my reading, the final "proof" of checkers only retains the results of these searches, but not the trees. At any rate, the $4 \cdot 10^{13}$ positions in 10-unit tablebases are much more useful than in Losing Chess, justifying both their large bulk and slow I/O constraints in searching.

[28] We could compare to orthodox chess where estimates are 10^{83} for the game-tree complexity and 10^{40} for the number of legal positions, though the forced-capture rule of Losing Chess and non-royal Kings respectively decrease/increase each of these numbers somewhat.

[29] If anything, our estimate of 1 million relevant pn-searches is probably high, and perhaps by a factor of 2 or more, though the long and ultimately unused work on 1. e3 b6 should also be noted here as roughly 50% of our effort.

[30] The wastage is ultimately due to our best-first expansion policy, which ends up following too many false/slow paths before switching.

[31] As one measure of this, recall that at early stage we decided to treat a draw as a "Black win" to remove such lines and speed things up.

8. REFERENCES

[A] L. V. Allis, M. van der Meulen, H. J. van den Herik, *Proof-number search.* Artificial Intelligence **66** (1994), 91–124. http://dx.doi.org/10.1016/0004-3702(94)90004-3

See also: L. V. Allis, *Searching for Solutions in Games and Artificial Intelligence.* Ph. D. Thesis, Department of Computer Science, University of Limburg & Universiteit Maastricht (1994).
http://pub.maastrichtuniversity.nl/36b5cf0a-cf06-4602-afdb-1af04d65c23b
http://fragrieu.free.fr/SearchingForSolutions.pdf

[BUH] D. Breuker, J. Uiterwijk, J. van den Herik, *The PN^2-search algorithm.* In J. van den Herik, B. Monien eds, Advances in Computer Games **9**, Universiteit Maastricht (2001), 115–132.
http://www.researchgate.net/publication/292699512_The_PN2-search_algorithm

See also D. Breuker, *Memory versus Search in Games.* Ph. D. Thesis, Department of Computer Science, Universiteit Maastricht (1998). http://breukerd.home.xs4all.nl/thesis

[ETC] A. Plaat, J. Schaeffer, W. Pijls, A. de Bruin, *Exploiting Graph Properties of Game Trees.* 13th National Conference on Artificial Intelligence (AAAI-96), Vol. 1, 234–239, ISBN 978-0-262-51091-2.
http://www.aaai.org/Papers/AAAI/1996/AAAI96-035.pdf

See also *Nearly Optimal Minimax Tree Search?*, Technical Report 94-19. Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.5551

[GHI] A. Kishimoto, M. Müller, *A General Solution to the Graph History Interaction Problem.* AAAI Proceedings of the 19th national conference on Artifical intelligence (2004), 644–649. ISBN:0-262-51183-5.
http://www.aaai.org/Papers/AAAI/2004/AAAI04-102.pdf

[K] A. Kishimoto, M. Winands, M. Müller, J.-T. Saito, *Game-Tree Search using Proof Numbers: The First Twenty Years.* ICGA Journal **35/3** (2012).
http://webdocs.cs.ualberta.ca/~mmueller/ps/ICGA2012PNS.pdf

[P] D. B. Pritchard, *The Classified Encyclopedia of Chess Variants.* (Second edition of *The Encyclopedia of Chess Variants*). Biddles Ltd., 2007. ISBN 978-0-9555168-0-1. Available from John Beasley.
http://www.jsbeasley.co.uk/encyc.htm

[S] J. Schaeffer, *Game Over: Black to Play and Draw in Checkers.* ICGA Journal **30** (2007), no. 4, 187–197.
http://icga.leidenuniv.nl/icga/journal/contents/Schaeffer07-01-08.pdf

[TB] J. Schaeffer, Y. Bjornsson, N. Burch, R. Lake, P. Lu, S. Sutphen, *Building the Checkers 10-piece Endgame Databases.* In *Advances in Computer Games* **10** (2003), edited by J. van den Herik, H. Iida, E. A. Heinz, Advances in Information and Communication Technology Vol. 135, Kluwer, 193–210. ISBN 1402077092.
http://dx.doi.org/10.1007/978-0-387-35706-5_13

[ST] Y. Shoham, S. Toledo, *Parallel Randomized Best-First Minimax Search.* Artificial Intelligence, **137** (2002), 165–196. http://dx.doi.org/10.1016/S0004-3702(02)00195-9

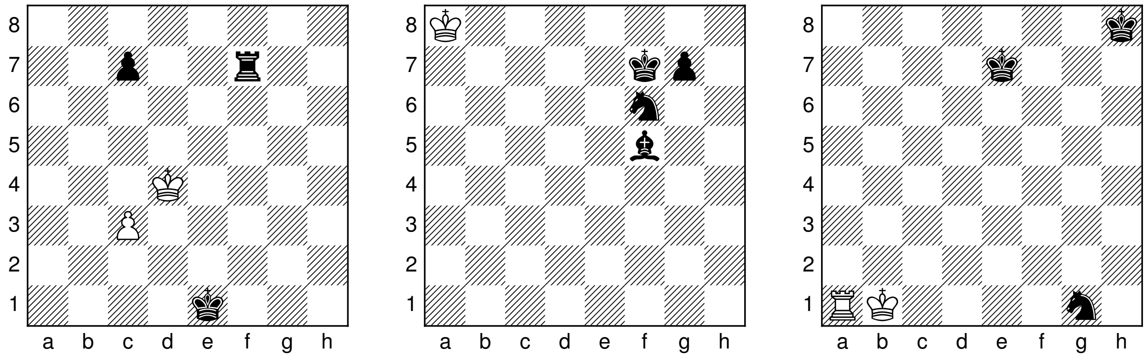
[V39] J. Beasley, *Losing Chess: The First Unofficial World Championship.* Variant Chess **39** (Jan 2002), page 106–107.
http://www.mayhematics.com/v/vol5/vc39.pdf

[V41] J. Beasley, *Losing Chess: Burning the Candle at Both Ends.* Variant Chess **41** (Jan 2003), page 8.
http://www.mayhematics.com/v/vol6/vc41.pdf

9. APPENDIX

Here we give some facts about tablebases. In our move sequences, we follow the standard notation that a single exclam denotes the sole move that minimaxes the length, while a double exclam is the only move (up to repetitions) that retains the game-theoretic result.

Figure 9: White to play and lose in 78 (left), in 74 under International Rules (center), and in 55 (right)



The longest line in 5-unit tablebases takes 78 moves to convert (Figure 9 left). Nye found the same (see [V41]).

1. Ke5! Rh7!! 2. c4! c5!! 3. Ke4! Rh1!! 4. Kf4! Kd1!! 5. Ke4! Kc1! 6. Kf3! Rd1 7. Kg2! Rd6 8. Kf1 Kb2! 9. Kf2! Ka3! 10. Ke2 Rh6! 11. Kd2 Rh8 12. Kd3 Ka4! 13. Ke4! Rh1! 14. Ke5! Ra1! 15. Kf5 Rd1! 16. Kg6! Rd2! 17. Kf6! Rd3! 18. Kf7! Ka3! 19. Kf6! Rd8! 20. Kg6! Kb2! 21. Kf5 Kc1 22. Kf4! Kd1! 23. Kg3! Rd6 24. Kg2! Kd2! 25. Kg3! Rb6! 26. Kg2! Rb2! 27. Kg3! Rb1! 28. Kg4! Ra1 29. Kg3! Rd1! 30. Kg4! Ke1! 31. Kf4! Kf1! 32. Kg4! Re1! 33. Kf5! Ke2! 34. Ke5! Rb1! 35. Kf5! Kf2! 36. Kg5! Re1 37. Kh5 Ke3! 38. Kg6! Rb1 39. Kf6 Kf3! 40. Kg6 Kg3! 41. Kg7! Ra1 42. Kg6 Rg1! 43. Kf6! Rh1! 44. Kg7! Kh4! 45. Kh7! Re1! 46. Kg7! Kg4! 47. Kf8! Rh1! 48. Kf7! Rh2! 49. Ke7 Kh5! 50. Ke6 Rh1 51. Ke5 Kh6! 52. Ke6! Kh7! 53. Ke5! Rg1! 54. Ke6! Rg8! 55. Ke5! Rg7! 56. Ke4! Kh6! 57. Ke5! Rh7! 58. Ke4! Kh5! 59. Ke3 Rh6 60. Ke4! Ra6 61. Ke3! Kg6! 62. Kf3! Kf6! 63. Ke3 Ke6! 64. Kf3! Rd6! 65. Kf2! Kf5! 66. Ke1! Rg6! 67. Ke2! Rg7 68. Kf1 Kg4! 69. Ke1! Rb7! 70. Kd2! Rb6 71. Ke1! Kf4! 72. Kd1! Rf6 73. Kc1 Re6 74. Kb1 Ke5! 75. Ka1 Re7 76. Ka2! Kd6! 77. Ka1! Re3! 78. Kb1 Kd5! 79. cxd5 c4!! 80. d6! c3!! 81. Ka2 c2!! 0-1

Another point of interest is that there is a KBNP vs K that takes 74 moves to convert under International Rules.

1. Kb7! Bh7!! 2. Kb6! Bb1!! 3. Kb5 Kg6!! 4. Kb4 Bf5!! 5. Kb5! Kg5! 6. Kb6! Bg6!! 7. Kb5! Kf4! 8. Kb4! Bf5!! 9. Kb5! Bg4! 10. Kb6! Ke5!! 11. Kb7! Bh5!! 12. Kb6! Kf5! 13. Kb5 Bg4!! 14. Kb4! Ke5 15. Kb5! Bh3! 16. Ka5! Bf5! 17. Kb4 Ng4 18. Kb3! Bc8!! 19. Kc2! Nf6! 20. Kb3! Bh3! 21. Kb2! Ng4! 22. Kc2! Kd5! 23. Kb3 Kd6! 24. Kc3 Nh6!! 25. Kd3 Nf7! 26. Ke2! Bd7! 27. Kd3! Ke6 28. Kc3! Bc8!! 29. Kd3! Ke7! 30. Ke3! Bd7! 31. Kd3! Kf8! 32. Ke3 Kg8! 33. Kd3 Bc8! 34. Ke3! Kh7! 35. Kf2! Kg6! 36. Kg2! Kf5!! 37. Kh3 Ke6!! 38. Kg3! Nd6! 39. Kf2 Bd7! 40. Kg2! Be8! 41. Kg3! Kf6! 42. Kf2 Ke5! 43. Ke2! Bd7!! 44. Kf2! Nf7! 45. Kg2! Be8! 46. Kh2! Nh8! 47. Kh3! Bf7! 48. Kh2! Ng6! 49. Kg2! Be8! 50. Kf2! Bd7! 51. Ke2! Bc8! 52. Kd2! Bb7! 53. Kc2! Kf4! 54. Kc3 Bg2 55. Kb4! Ke5! 56. Kb5! Bh3!! 57. Kb4! Bg4! 58. Kb3! Nf4! 59. Kb2! Bc8! 60. Kc1! Nh3! 61. Kd2 Bb7! 62. Kc2! Kf4! 63. Kc3 g5! 64. Kc4! Bc8!! 65. Kc5 Bf5! 66. Kc4! Bg4! 67. Kc5! Bd1! 68. Kd6 Kf3! 69. Ke6 Ke3 70. Kd6 g4! 71. Kc5 Kf3! 72. Kd6 g3! 73. Ke6 g2! 74. Kd6 g1=R (win in 7)

However, in joint Rules, the original position is drawn. In the above line, White plays 35. Kf3!! (Kf2 loses in 54), and Black is forced to accept a draw after 35... Bxg4 36. Kg4 Nd8 since 36... Ng5 37. Kxg5 Kg6 38. Kxg6 is only drawn. The longest conversion for KBNP vs K in joint Rules is 67 moves (wKa1 bKe5 bBg4 bNg6 bPc7).

Both KRN vs KN and KKN vs KR have maximal paths of length 55 in joint Rules, and we give the latter.

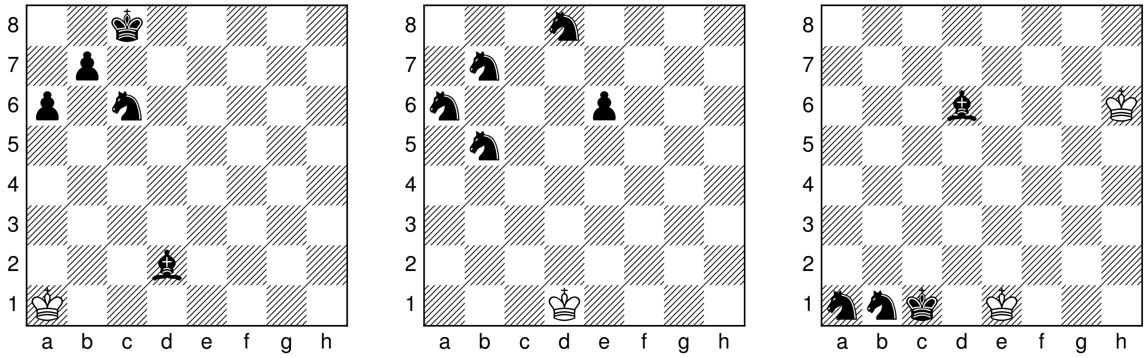
1. Kc1! Nh3 2. Kd1! Kf6!! 3. Rc1! Kf5!! 4. Ke1! Ng5!! 5. Ke2! Nh7!! 6. Kd3! Kg7!! 7. Rd1! Ke6!! 8. Rc1! Kef6 9. Rc3! Kgg6 10. Ke3! Kh5!! 11. Rd3! Kh4 12. Rd1! Kg6! 13. Kd4! Nf8!! 14. Ra1 Kg3 15. Rd1! Kg4! 16. Rd2! Kh4! 17. Rc2! Khg5! 18. Rc4! Kf7! 19. Rc3! Kff6!! 20. Kc4! Kgf5! 21. Kc5! Kf7! 22. Kc4! K7g6! 23. Rc1 Kg4! 24. Kb4 Kf7! 25. Kc5! Kf6! 26. Rc2! Ng6! 27. Kd4! Kgf5! 28. Kc5! K6g5! 29. Rc3! Kfg4! 30. Rc1! K4f4 31. Kc4! Kf3! 32. Kc5! Kf6! 33. Kc6! Nh4! 34. Kb5! Nf5 35. Rc8 Ke3 36. Kb4! Ng3 37. Ka3! Kf4 38. Kb2 Nh5 39. Kc3! Ke6! 40. Kb3! Nf6 41. Rc1! Ke7! 42. Rc2! Ke5! 43. Rc6! Ne8!! 44. Ra6! Ke4! 45. Ra1 Kd6 46. Rb1 Nf6 47. Rb2! Kde5 48. Rb1 K5f4! 49. Ra1 Kee3 50. Kb2! Kf2 51. Kc1! Kf1 52. Ra7! Kg2! 53. Ra1! Kh1! 54. Rb1! Ke3! 55. Ra1 Kd2 56. Kxd2 Ng4 57. Rxh1 Nh2 58. Rxh2# 0-1

Comparatively, there is already a pawnless 4-unit TB (wKd1 bKa3 bBf6 bNa5) that takes 71 moves to convert.

9.1 6-unit TBs

Subject to the caveats in the main text about these tablebases, we give some maximal lines.

Figure 10: White to play and lose in 93 (left), in 89 (center), and in 86 (right)



1. Ka2! Bg5!! 2. Kb2! Bh4 3. Kc1 Be7!! 4. Kd1 Nb8! 5. Kc1 Kc7 6. Kb2 Kd6!! 7. Ka3 Nd7 8. Kb3! Bh4! 9. Kc3! Ke6! 10. Kb3! Ke5! 11. Kc2! Kd5! 12. Kd2! Be7!! 13. Ke3! Kc6!! 14. Kf3! Bf8! 15. Kg4! Nb6! 16. Kg5! Bb4!! 17. Kf6! Ba5!! 18. Kf5! Na4! 19. Kf4 Bb4! 20. Kf5! Ba3 21. Kg6! Bc5! 22. Kf5! Bb4! 23. Kf4! Kd7! 24. Ke3! Ba3! 25. Ke4! Bf8! 26. Kf3 Nb6! 27. Kg4! Bb4! 28. Kf4! Ba3! 29. Ke3! Ke6! 30. Kf3! Bb4 31. Ke2 Be7! 32. Kf3! Ba3! 33. Ke3! Bf8! 34. Kf3! Kd6! 35. Kg3 Kd5! 36. Kg4 Na8! 37. Kg5 Bc5! 38. Kh6! Kd6! 39. Kg6! Nb6! 40. Kg7! Bg1! 41. Kg6! Na4! 42. Kf7! Kc5!! 43. Kg6! Nc3! 44. Kf6! Kc6! 45. Kg5! Kd6! 46. Kg4 Bc5!! 47. Kg3 Ba3!! 48. Kg4 Nb5 49. Kg5 Bb4! 50. Kg6! Ba5! 51. Kg5 Kd5! 52. Kg4! Bb4! 53. Kg5! Nc7! 54. Kg6 Kc6! 55. Kg7! Ba5! 56. Kg6 Bb6! 57. Kg7! Bf2 58. Kg6! Bg1 59. Kg7! Bb6! 60. Kg6 Kc5! 61. Kg5! Kd4 62. Kg4 Bc5!! 63. Kg5! Kc4 64. Kf4! Ba3 65. Kf3 a5 66. Kf4! Bf8 67. Kg5! Bb4! 68. Kg6! Bc5! 69. Kg5 Na6! 70. Kf6! Bg1 71. Kg5! Ba7! 72. Kg4 Bc5! 73. Kg3 Ba3! 74. Kf2 a4 75. Ke2 Kb3!! 76. Ke3! Nc7! 77. Ke2 Be7! 78. Kd1! Kc4! 79. Ke2! Kb4! 80. Kd1 Ne6! 81. Ke2! Nd8! 82. Kd1 b6! 83. Ke2! Nb7! 84. Ke1 Na5! 85. Ke2! Bd6 86. Kd1 Kc4! 87. Kc1 Bg3 88. Kb1 b5! 89. Kc1! Nb7! 90. Kd1! Be5! 91. Ke1! a3! 92. Kd1 a2!! 93. Kc1 Bb2!! 94. Kxb2 0-1

Here is a four Knights win, where after Black's 51st move the Knights have retreated to c8/d8/e8/f8.

1. Ke2! N5d6!! 2. Kf2! N8f7!! 3. Kg3! Ne8!! 4. Kf3! Nbd8!! 5. Ke3! Nh8! 6. Kd3 Nac7!! 7. Ke3! Ndf7! 8. Kd3! Nh6!! 9. Ke3! Ng8 10. Kf3! Ng7!! 11. Ke3! Na6! 12. Kf3! Nf7! 13. Ke3! Ne8 14. Kd3 Ne7! 15. Kc3 Nac7!! 16. Kd3! Nh6! 17. Ke3! Nhg8! 18. Kd3! Ng7! 19. Kc3! Nf6! 20. Kd3! Nh7! 21. Kc3 Nge8! 22. Kd3! Nf8! 23. Ke3! Na6 24. Kf3! Nh7!! 25. Ke3! Nb8! 26. Kd3! Nd7! 27. Ke3! Nc8 28. Kd3 Nc7! 29. Ke3! Na8! 30. Kd3! Nhf8!! 31. Ke3! Nab6 32. Kf3! Nh7!! 33. Ke3! Ne7! 34. Kd3! Nbc8!! 35. Ke3! Ng8! 36. Kf3! Nce7! 37. Ke3! Nh6! 38. Kd3! Nc8! 39. Ke3! Na7! 40. Kd3! Ndf8! 41. Ke3! Nf7! 42. Kd3 Nd8!! 43. Kc3 Nd7! 44. Kd3 Nc8!! 45. Ke3! Nb8! 46. Kf3! Ne7!! 47. Ke3! Na6! 48. Kd2 Nc7 49. Kc2 Nc8 50. Kd3! Nf8! 51. Ke3! Ne8! 52. Kd3! Na7! 53. Kc3! Nb7 54. Kd3! Ng6! 55. Ke3! Ne7!! 56. Kd3 Nac8 57. Ke3! Ng7! 58. Kf3! Nd8! 59. Ke3! Nf7! 60. Kd3 Na7!! 61. Ke3! Nh6! 62. Kf3! Nhg8! 63. Kg3! Nac8! 64. Kf3! Nb6!! 65. Ke3! Nec8! 66. Kd3 Nd7!! 67. Ke3! Na7 68. Ke2 Nc6 69. Kf2 Nb6 70. Kf3! Nge7! 71. Kg3! Nc4 72. Kg2! Nd6! 73. Kg3! Nb8! 74. Kf2! Nd7! 75. Ke2 Nc6! 76. Kd2 Nf8! 77. Ke2! Ng6! 78. Kf2! Nb4! 79. Kg2! e5! 80. Kf1 Nd5 81. Kg1! Nb7 82. Kf1 Nc5 83. Ke1 Nge6! 84. Kf1! Ng5! 85. Kg1! Nb3 86. Kf1! Nb4 87. Kg1! Nd4! 88. Kh1! Nd3 89. Kg1 Nf2! 90. Kxf2 0-1

And finally we give the longest pawnless example.

1. Kf2! Kd1!! 2. Ke3 Ba3!! 3. Kf4! Nc2!! 4. Kf5! Bc5!! 5. Ke6! Bg1!! 6. Kg5 Na1!! 7. Kgf6! Na3!! 8. Kef5! Ke2!! 9. K6g5 Nb1!! 10. Kgf6 Ba7! 11. K6g6! Kd2! 12. Kgg5! Kd1! 13. Kff6 Kc2! 14. Kgf5! Na3! 15. K6g5 Bb6! 16. Ke6! Kd3! 17. Kgf6! Bg1 18. Kff7! N1c2! 19. Kg6! Ne1! 20. Kgf6! Nb1 21. Kfe7! Kc3! 22. K7f6! Kb3! 23. Kff5! Ba7! 24. Kg5 Kc4! 25. Kef5! Na3 26. Kgg6! Bb6! 27. Kff6 Nec2! 28. Ke7! Bg1 29. Kf5! Na1! 30. Kf7! Bf2 31. Kg4 Bb6! 32. Ke7 Ba7!! 33. Kg3 Nb3 34. Kh4! Nc1 35. Kg4! Kc3 36. Ke6! Kd3! 37. Kg5! Kc4! 38. Ke7! Kd4! 39. Kg6 Ne2! 40. Kgf6 Kc4! 41. Kg6 Nb1 42. Kgf6! Nd2 43. Kd7 Bg1! 44. Kde7! Kb5! 45. Kef7 Kc5! 46. K7g7! Nb1 47. Kgg6! Nbc3 48. Kgg7! Kc4! 49. Ke7! Kd4! 50. Kg6! Kc5! 51. Kef6 Kc4! 52. Ke7! Kd4! 53. Kgf7! Ng3 54. Kd7 Nf1! 55. Kc7 Ne3 56. Ke7! Bf2! 57. Kc8 Ncd1 58. Ked7 Be1! 59. Kb7 Nc2 60. Ke7! Nf2! 61. Kd7! Bc3! 62. Kbc8! Nd3 63. Kcd8! Kc4! 64. K8e8! Ne3 65. Kf8! Kb4! 66. Kfe8! Be1! 67. Kf7! Bf2! 68. Kc7! Kb5! 69. Kc8! Kc5! 70. Kd8! Nd1 71. Kg6 Bg1 72. Kg5 N1b2 73. Kg6! Kd5! 74. Ke8! Nc4 75. Kf8! Nb4 76. Kg5 Bh2 77. Kg6! Nd2 78. Ke8! Bb8 79. Kf8! Ke4! 80. Kh6! Ke5! 81. Kh7! Nf3 82. Khg7 Kd5! 83. Kh7 Bh2 84. Kfg7! Nc6 85. Kgg8 Be5! 86. Kg6 Ng5 87. Kxg5 0-1

9.2 Zugzwangs

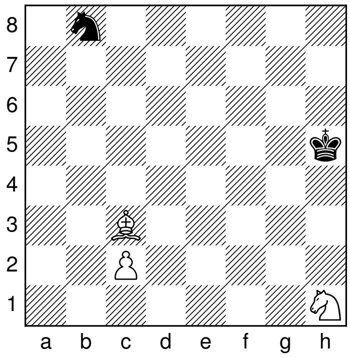
We can also list some full-point zugzwangs[32] of interest. Already in the 4-unit TBs we have wKc5 bKa2 bKa1 bNb1 (Kkkn 45/2) as one where White takes 45 moves to lose (Black loses in 2, the idea here being that only a Kb2 move avoids immediate loss, but then White can play Kd4, and Black has c3 doubly attacked). With NNnn there is the symmetrical wNh8 wNa4 bNa1 bNh5 (NNnn 36/36) that loses in 36 for whomever is on move, and others such as wNg1 wNh2 bNa8 bNb7 (NNnn 31/23); indeed, often in Losing Chess one eliminates Knights and pawns from zugzwang accounting, as they tend to create too many. This leaves Rkkk with wRd3 bKb6 bKb1 bKg1 (Rkkk 20/7) as the longest at 20 moves, and if one excludes this grouping, then there is nothing more than 7 moves, here wRa5 wKf3 wKc2 wBd1 (Rkkb 7/2) and wKd6 bQb1 bRf2 bBg1 (Kqrb 7/2). The symmetrical wKc2 wBb1 bKf7 bKg8 (KBkb 5/5) is also perhaps worth mentioning, as is wRa8 wPa2 bRe1 bPe7 (RPrp 8/8) and variants. Another example with pawns is wQa4 wPb4 bBh4 bPh6 (QPbp 7/12), and one without is wKe7 wKa5 bQb2 bQf1 (KKqq 3/3).

A list of all zugzwangs is available from the main project website, though it must be warned that some data might be partially redundant, due to various symmetries (including shifting).

9.2.1 3-vs-2

The situation is similar to the 4-unit genre. When allowing Knights and pawns, there is a loss in 42 from wNc4 wPg4 bNh8 bNb8 bPg5 (NPnnp 42/3), and a wider piece variety for wNe4 wNg1 bRa6 bNa1 bPg7 (NNrnp 34/5). The most notable example where both sides have a significant number of moves to make is wBc3 wNh1 wPc2 bKh5 bNb8 (BNPkn 19/15, Figure 11).

Figure 11: White loses in 19, Black in 15

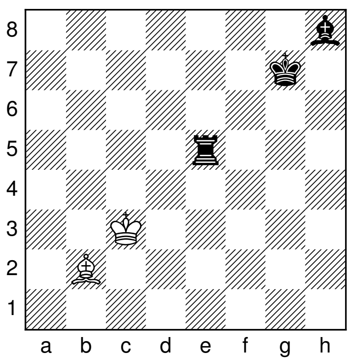


1. Ba1 Na6!! 2. Nf2! Kg6 3. Bb2 Nc7! 4. Ba3! Kf5 5. Bc5! Ke6! 6. Be3 Ne8 7. Nd3! Ke7 8. Nb4! Kf6! 9. Bg1 Kg5! 10. Ba7! Ng7! 11. Bg1 Nh5! 12. Bb6! Kg4 13. Ba5! Ng3 14. c3 Nf5 15. c4! Kg5 16. Nd3! Kf6 17. Nf2! Ne7 18. c5! Nc8! 19. c6 Nb6! 0-1

1... Kf4! 2. Bb4!! Kf5! 3. Nf2!! Ke6! 4. Bf8! Kf7! 5. Ba3! Kf6! 6. Bc5! Ke6! 7. Nd1 Kf5! 8. Nb2 Ke6! 9. Ba3 Ke5! 10. Bb4 Ke6! 11. Bc5! Kf5 12. c4! Kf6! 13. Bb4 Kf7 14. c5! Kf8! 15. Be1 Nc6 16. Ba5 1-0

Upon excluding Knights and pawns from the mix of pieces, we have losses in 8 for wKc1 wKb2 wBa1 bRe7 bBf8 (KKBrb 8/1), wKc2 wKb3 wBb1 bKb5 bRe4 (KKBkr 8/1), and wKd8 wKc2 wBb1 bQh4 bRf6 (KKBqr 8/1). There is also wKc3 wBb2 bKg7 bRe5 bBh8 (KBkrb 6/4, Figure 12 left) and wKg8 wKh6 bQd1 bRe2 bBh4 (KKqrb 6/4, Figure 12 right) where both sides have a few moves to make before losing.

Figure 12: Both positions: White loses in 6, Black in 4



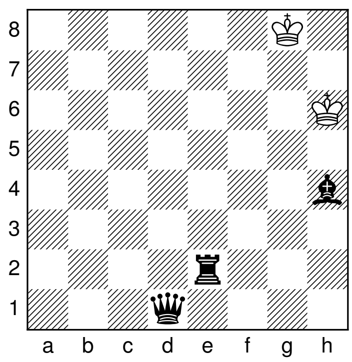
1. Ba1! Rg5 2. Bb2! Kf6!! 3. Ba1! Bg7! 4. Bb2! Bh6! 5. Ba1! Bf8 6. Bb2 Bb4 0-1

1... Kf6! 2. Bc1 Rh5! 3. Ba3!! Rh7! 4. Kd4 Kg7 5. Bf8 1-0

---

1. Kgh7! Bg3!! 2. K6g6! Bd6! 3. Kh5! Qf1!! 4. K5h6! Bh2 5. Kh5 Re6! 0-1

1... Be1 2. Kh5! Bf2 3. Kg7 Bg1 4. Kf6 Ba7 5. Ke5 1-0

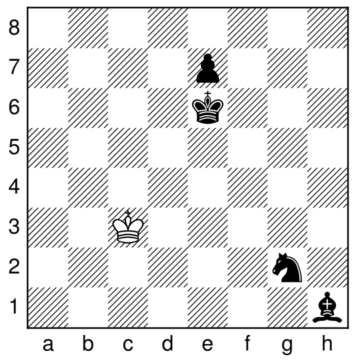


[32] This is a piece configuration where White to move has Black winning with best play, and vice-versa.

9.2.2 4-vs-1

The longest-lost zugzwang is with wKc2 bRg4 bBh5 bNh6 bPh7 (Krbnp 43/2) where White to move takes 43 moves for Black to convert.[33] The losses with long play for both sides include wNc4 wPb4 wPa3 wPc2 bNe7 (NPPPn 14/13) and the more interesting (in my opinion) wKc3 bKe6 bBh1 bNg2 bPe7 (Kkbnp 33/10, Figure 13).

Figure 13: White loses in 33, Black in 10

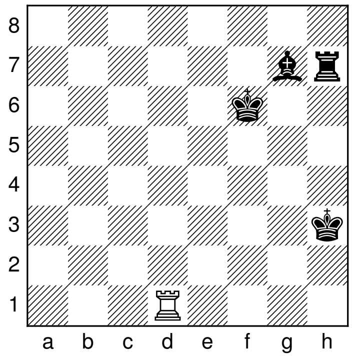


1. Kc2! Nh4!! 2. Kc3! Kf6! 3. Kd2! Ng6!! 4. Ke1! Bd5! 5. Kf1! Ba2 6. Ke1 Ke5! 7. Ke2! Nf8 8. Kd1 Bd5 9. Ke1! Nd7 10. Kf1! Bf7 11. Ke1 Nc5 12. Kf2! Ba2! 13. Kg2! Na6 14. Kf1 Nb4 15. Kg2! e6 16. Kh2 Bb1! 17. Kg2! Nc6 18. Kf2! Ne7! 19. Kg2! Nc8! 20. Kf2! Nd6! 21. Kg1 Nc4! 22. Kh2! Bh7 23. Kg2! Kf5! 24. Kg1 Ke4 25. Kf1 e5 26. Kg1! Na5 27. Kh1 Nb3 28. Kh2! Bg6! 29. Kg1! Bf5! 30. Kf1 Be6 31. Kg1! Nd4 32. Kh1! Ke3! 33. Kg1 Kf2! 34. Kxf2 0-1

1... Kd6! 2. Kb4!! Ke6! 3. Kb5! Ke5! 4. Kb6! Ke4 5. Kc7! e5! 6. Kb6!! Kf5! 7. Kc5! e4! 8. Kb4 e3! 9. Kc3! e2! 10. Kc2!! e1=Q 11. Kd1 1-0

Upon excluding Knights and pawns, we have a 13-move loss for White in wKd4 bKg3 bRg6 bBh7 bBh2 (Kkrbb 13/1), and extended losses for both sides in wRc2 bKe7 bKg4 bRg8 bBf8 (Rkkrb 7/4) and wRd1 bKf6 bKh3 bRh7 bBg7 (Rkkrb 7/4, Figure 14 left), and wRd1 bKh8 bKf7 bKg3 bRg7 (Rkkkr 6/4, Figure 14 right), where the last pieces can be moved to bKh3 bRh7 with a similar solution.

Figure 14: Left: White loses in 7, Black in 4. Right: White loses in 6, Black in 4.



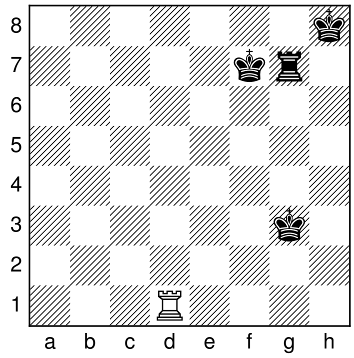
1. Rb1! Ke5!! 2. Rc1! Rh5!! 3. Rb1! Bf6! 4. Rc1! Rh6! 5. Rb1! Kd4! 6. Ra1! Kc3 7. Ra8 Bd8!! 0-1

1... Kh4! 2. Rd2!! Kh5! 3. Rd3! Kh6 4. Rg3 Bh8 5. Rg5 1-0

---

1. Ra1 Ke7! 2. Rc1! Kh7 3. Ra1! Kd7! 4. Rb1! Kh8 5. Ra1! Kc7! 0-1

1... Kg4! 2. Rd2!! Kg5! 3. Rd3! Kh7 4. Rg3! 1-0 (or Rd7 in similar lines)



I have not yet attempted to calculate zugzwangs in the limited 6-piece data, partially due to incompleteness.

[33] Under International Rules there is wKc2 bQh4 bNf4 bNg3 bPc6 where White takes 50 moves to lose, but this is drawn under joint Rules: upon reaching wKe2 bNa6 bPc6 bNd6 bQh8 in 24 moves in the mainline, White has Kf3 Ne4, Kxe4 Qd4, Kxd4 Nc5, with a FICS draw.