

Reach Chess: An accidental chess variant

Cameron Browne^{a,b,*}, Markus Niebisch^{b,c}, Alain Brobecker^a and Stephen Tavener^a

^a *Independent*

^b *Department of Advanced Computing Sciences, Maastricht University, The Netherlands*

^c *Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands*

Abstract. Reach Chess is a chess variant discovered by accident while testing the Ludii general game system, in which a simple rule change fundamentally subverts the nature of the standard game. This paper describes Reach Chess, how it came to exist, and reveals its character through a brief analysis and puzzle with annotated solution.

1. INTRODUCTION

Reach Chess is a simple chess variant played as per standard chess except that:

1. A player wins by reaching the far rank with any piece of their colour.
2. A player cannot make any move which would leave their king in check.
3. A player with no legal moves must pass that turn.¹

Figure 1 (left) shows the Reach Chess starting position using standard chess equipment and initial setup, and (right) shows a game won by White who has occupied the far rank 8 with a white rook.

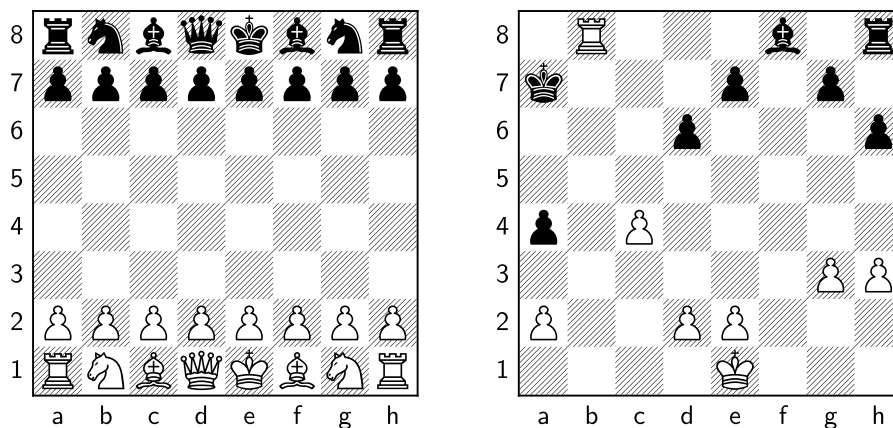


Fig. 1. Reach chess start position (left) and a game won by white (right).

The game ends when a winning move is made, so it does not matter that the winning rook shown in Fig. 1 (right) is subsequently attacked by the black king. Reach Chess is a fast and attacking game, as indicated by the fact that most of the stronger pieces have been consumed in the example shown.

*Corresponding author. E-mails: cambolbro@gmail.com, cameron.browne@maastrichtuniversity.nl.

¹If neither player has any legal moves, then the game is declared a draw. Such “double stalemate” situations will be rare.

It is worth clarifying the second rule, that *a player cannot make any move which would leave their king in check*. This implies that: a) a player whose king is in check on their turn must play a move out of check if possible, and b) a player cannot make any move that would put their king in check (e.g. moving to reveal an enemy attack). This takes precedence over a winning move.

For example, Fig. 2 shows a situation in which White has a potentially winning move to c8 but is also in check (left). White must move out of check with highest priority, allowing Black to then win next turn with e1 (right).

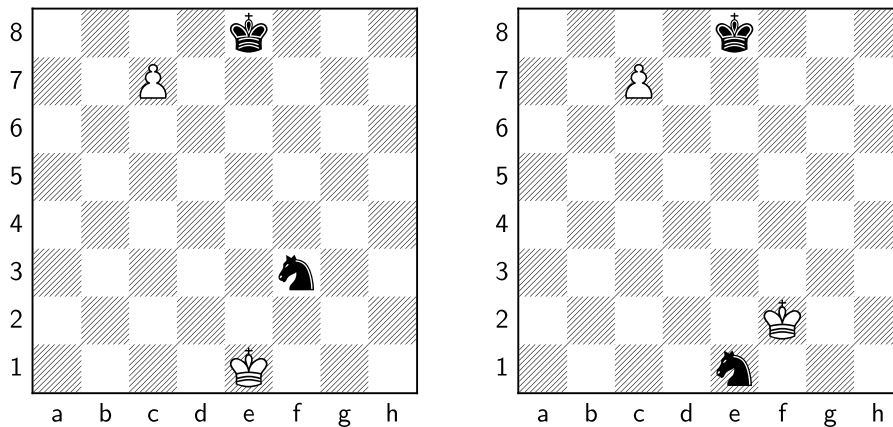


Fig. 2. White (to play) must move out of check, allowing black to win (right).

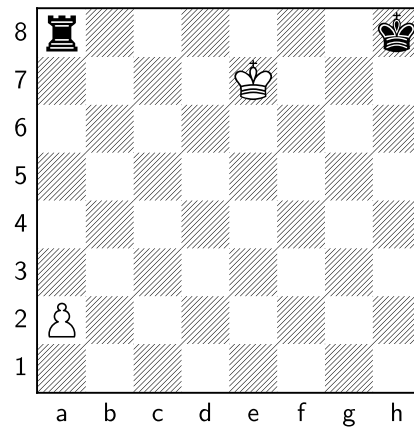


Fig. 3. White cannot move into check to win.

Similarly, a player cannot move into check even if it that move would reach the far rank. Figure 3 shows a position in which White has potentially winning moves with their king to d8, e8 and f8, but all would move the king into check so cannot be made. Black should win from this position.

2. ORIGINS

Reach Chess was discovered in February 2021 by the second author (Markus Niebisch) while working on the Ludii general game system (Browne, 2019) as part of a Masters Research Internship at

Maastricht University’s Department of Advanced Computing Sciences (DACS). This work involved evolving games described in the Ludii game description language (Browne et al., 2020) using standard *genetic programming* operators of *mutation* and *crossover* (Koza, 1992).

This approach for evolving new games from existing descriptions had proven successful for the earlier Ludi general game system (Browne, 2011) and the task was now to test it on the recently devised Ludii language. The Ludi and Ludii languages are similar in nature – both describe games as structured sets of *ludemes* (Browne, 2021) – but the Ludii language is more comprehensive and complex.

To test the crossover operator, compatible ludemes were swapped between several pairs of standard board games, which fortuitously included chess and Hex.² The chess end condition stipulates that a player wins by checkmating the opponent’s king or draws if there are no legal moves (i.e. stalemate) or 100 moves are made without a capture or pawn move, as follows:

```
(end {
  ("Checkmate" "King")
  (if
    (or (no Moves Mover) (= (counter) 99))
    (result Mover Draw))
  )
})
```

This ludeme in the chess description was replaced by the following Hex end condition, while the rest of the chess description remained unchanged:

```
(end (if (is Connected Mover) (result Mover Win)))
```

Random self-play trials quickly revealed the resulting games to be reasonably balanced, then manual testing confirmed this new rule set to be an interesting game in its own right. With this simple rule swap, Reach Chess was born.

3. WHY IT WORKS

It was a surprise to all involved that this simple modification produced the behaviour it did. Specifically, *why should a connection test detect when a piece reaches a certain position?* The answer to this question lies in the way in which connection testing is implemented in the Ludii game system.

Firstly, it is important to note that each player’s promotion zone on the far rank of the board is explicitly labelled in the ludemic chess description as belonging to its owner, either White (P1) or Black (P2), as follows:

```
(regions "Promotion" P1 (sites Top))
(regions "Promotion" P2 (sites Bottom))
```

Connection games in the Ludii language require target regions to be defined for each player, so that the `(is Connected Mover)` ludeme can determine whether all regions belonging to the mover are connected by a contiguous set of their pieces. In the case of Reach Chess, only a single region

²The aim in Hex is to connect the opposite sides of the (hexagonally tessellated) board marked your colour with a connected set of pieces of your colour.

is defined for each player – which happens to be the far rank – and this region is deemed to be “connected” when any piece belonging to its owner lands there. The fact that these regions were intended to define promotion zones is inconsequential.

The “must move out of check” and “must pass if checkmated” rules of Reach Chess are additional side-effects of the ludemic description, this time due to the following clause – appended to all moves – which filters out those potential moves that would leave the player’s king in check. If no legal moves remain, then the default behaviour is to pass:

```
ifAfterwards: (not ("IsInCheck" "King" Mover))
```

We describe these surprising behaviours as “quirks” of the program rather than bugs, as the code works correctly in expected cases and has proven surprisingly robust to this unexpected case, yielding interesting results where none could reasonably be expected.

4. NOVELTY

Is Reach Chess a new game? It is hard to believe that this simple modification has not been tried before, but no chess variant with exactly the same rules has been found so far in the literature.

Pritchard’s revised encyclopedia of chess variants lists some similar games in its Section 10.7 “Reaching a fixed goal” (Pritchard, 2007, pp. 84–85). These include Bishop Chess, in which the aim is to move a bishop to the far rank, and Outpost Chess, in which players nominate an “outpost” cell and the aim is to occupy the opponent’s outpost for a turn.

King of the Hill³ is another well-known chess variant played as per standard chess with the additional rule that a player also wins by occupying any of the four central cells with their king. Échecs Football (Football Chess)⁴ is a chess variant in which White wins by installing any white piece on cells d8 or e8 and Black wins by installing any black piece on cells d1 or e1.

Chess variants similar to Reach Chess typically retain checkmate as a way to win the game, or only require a certain subset of pieces to reach a certain subset of cells. Retaining checkmate can create some emergent tension; for example, in King of the Hill players must be careful of exposing their king to checkmate as it approaches the central cells, while conversely a tempting sequence of checking moves may actually chase the opponent’s king closer to the central target cells.

5. ANALYSIS

Since a player who cannot move out of check must pass (rule 3), that player will almost certainly lose the game. Such situations are *effective checkmates* even though the game does not have a checkmate rule as such. Even so, the “reach the far rank” aspect of the game (rule 1) supercedes the material aspect and position becomes more important; even powerful pieces can be reduced to mere blockers to impede the opponent’s line-of-sight. Swapping a queen for a pawn can be a good move if that opens a vital line to the far rank.

This equalising of piece importance is exacerbated by the fact that the game ends as soon as a winning move is made, hence players do not need to plan for the survival of their pieces beyond the winning

³<https://www.chess.com/terms/king-of-the-hill>

⁴<http://abrobecker.free.fr/chess/fairyblitz.htm#football>

strike. For example, it is irrelevant that the white rook in Fig. 1 (right) is directly attacked after its winning move.

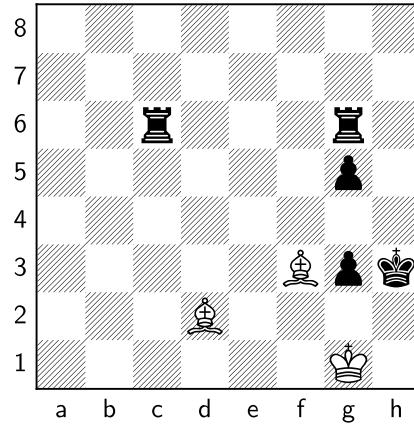


Fig. 4. White to play and win.

Reach Chess is a fast and attacking game in which players can exchange pieces with abandon. The line from any piece to the far rank is rarely through more than a couple of the opponent's pieces, which maintains a high level of tension throughout each game. And while positional play becomes more important, it is the position of line-of-sight that matters more than the position of threat and defence. Each game can be seen as a traffic jam in which players strive to clear a passage to the far side of the board before their opponent does.

Figure 4 shows a Reach Chess puzzle that demonstrates these principles in action. White (to play) appears to be in a hopeless position, but can turn the game by reducing the more powerful black rooks into simple blockers and pinning them in place. A complete annotated solution is provided in the Appendix.

6. COMPARISON

Reach Chess was implemented in the Ai Ai general game system (Tavener, 2015–2024) and automated self-play analyses were performed to compare its behaviour to standard chess. The chess analysis was made over 1,055 self-play games and the Reach Chess analysis was made over 2,775 self-play games, all using standard Alpha Beta search with 1s thinking time per move (around 6-ply search on average).

Table 1 shows some selected results that highlight differences between the two games. The full reports can be found on the Ai Ai reports page.⁵

Table 1
Selected results from Ai Ai analyses of chess and reach chess

Game	First Move Wins	Draws	Game Length	Surprise
Chess	54.22 ±3.02%	18.58%	≈ 128.92 moves	21.71%
Reach Chess	51.98 ±1.86%	1.91%	≈ 35.81 moves	58.33%

⁵The full Ai Ai reports for Chess and Reach Chess (and many other games) can be found at: <http://mrraow.com/uploads/AiAiReports/reportsIndex.html>.

These results suggest the following key findings:

- **Reach Chess is more balanced than chess.** Reach Chess has around half the observed first-move advantage for White, which is surprising given that it is the more attacking game.
- **Reach Chess has fewer draws than chess.** Even the few draws reported here are probably due to the Ai Ai implementation bending the Reach Chess rules slightly to include standard chess checkmate and stalemate, in order to use its existing chess code as much as possible (which does not affect the game unduly). Draws in actual Reach Chess games will only be due to “double stalemate” situations, which will be extremely rare in practice.
- **Reach Chess games are shorter than chess.** Reach Chess games are about a quarter as long as chess games on average. Note that average game lengths tend to be longer for AI self-play trials than for actual games between human players, as AI agents will play on to the bitter end, even from weak positions in which human players would typically resign.
- **Reach Chess has a higher incidence of “surprising” moves.** These are moves for which shallow search gives different value estimates than deeper search, i.e. moves which initially look good to the Ai Ai agent but prove to be not so good on deeper analysis (and vice versa). This may be because the agent uses heuristics fine-tuned for chess, which fail to capture some important Reach Chess concepts and misdirect the early stages of the search.

7. CONCLUSION

Reach Chess is a simple chess variant discovered during routine testing of the Ludii general game system that stands as a game in its own right. It is similar to standard chess – a strong chess player will generally beat a weaker one at Reach Chess – but has its own distinct character, as seen in the following puzzle analysis. This fundamental change from a single rule swap is a rare example of (pleasantly surprising) emergent behaviour arising from unexpected input.

This new game of Reach Chess is very much a discovery in the Platonist sense rather than a deliberate invention (Balaguer, 2016), and highlights the importance of considering *every* candidate, regardless of apparent plausibility, when automatically searching for new and interesting games (Browne, 2011).

ACKNOWLEDGEMENTS

This work was partly funded by the European Research Council through ERC Consolidator Grant #771292) as part of the Digital Ludeme Project (Browne, 2019). Thanks to I. M. Benjamin Defromont, F. M. Sven Charmeteau, Axel Gilbert and Jacques Dupin for composing the example puzzle and providing analysis. Stephen Tavener implemented Reach Chess in Ai Ai and prepared the reports used for the Comparison section.

APPENDIX. PUZZLE SOLUTION

This appendix annotates a solution for the Reach Chess puzzle shown in Fig. 4.

The Black rook c6 threatens to reach c1 to win, so White interposes a bishop to block it (capturing the rook would be a losing play):

1. Bd2-c3

White now threatens to win with 2. Bc3-h8#. Black cannot capture with 1... Rc6 × c3 as 2. Bf3-a8# then wins, so must block this threat:

1... Rg6-f6

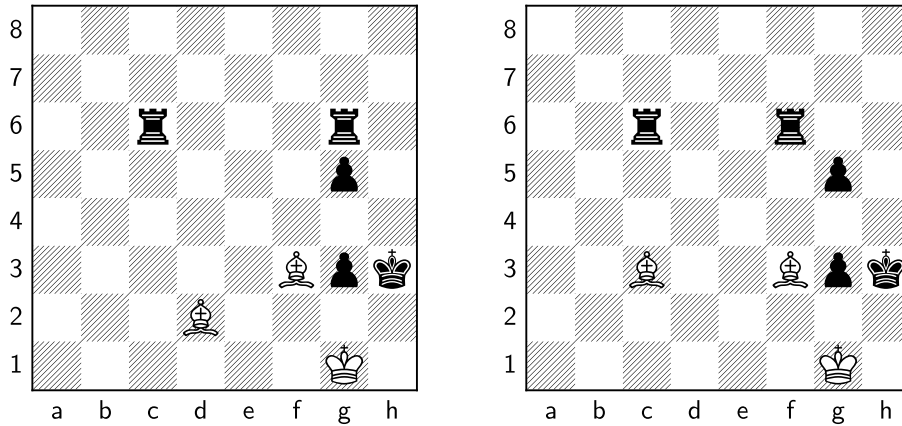


Fig. 5. White to play (left) and position after 1. Bd2-c3 and Rg6-f6 (right).

The quartet made by the bishops and rooks are now in a delicate blocking balance, and moving any of these is potentially disastrous. White moves their king away from the action and endeavours to put Black in *zugzwang*:

2. Kg1-h1

Black now has only three options that do not disturb their rooks. One is 2... Kh3-h4? 3. Kh1-g2 which leads to a *zugzwang* position one move sooner. Another (losing) option is: 2... g5-g4? 3. Bf3-g2+ Kh3-h4 4. Bc3 × f6+ Rc6 × f6 5. Bg2-a8#. So black chooses the remaining option:

2... g3-g2+

White continues to play safe without unduly disturbing the position:

3. Kh1-g1

Black could now make the mistake 3... Kh3-g3? allowing the fork 4. Bc3-e5+. The better move is:

3... Kh3-h4

White now eliminates all movement options for the black king:

4. Kg1 × g2 (*zugzwang*)

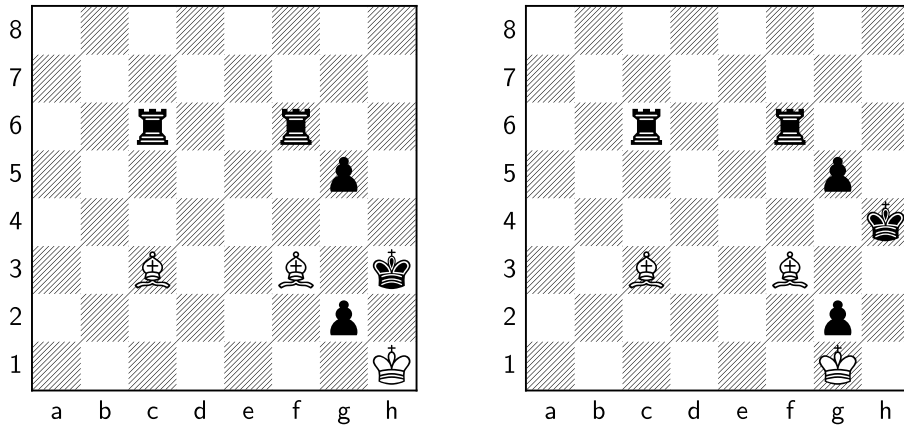


Fig. 6. Positions after move pairs 2 (left) and 3 (right).

Moving a rook without making check is still bad for Black (e.g. 4... Rc6 \times c3 5. Bf3-a8# or 4... Rf6 \times f3 5. Bc3-h8#), so they must advance their pawn:

4.... g5-g4

White captures the rook and puts Black in check:

5. Bc3 \times f6+

Black recaptures to get out of check:

5... Rc3 \times f6

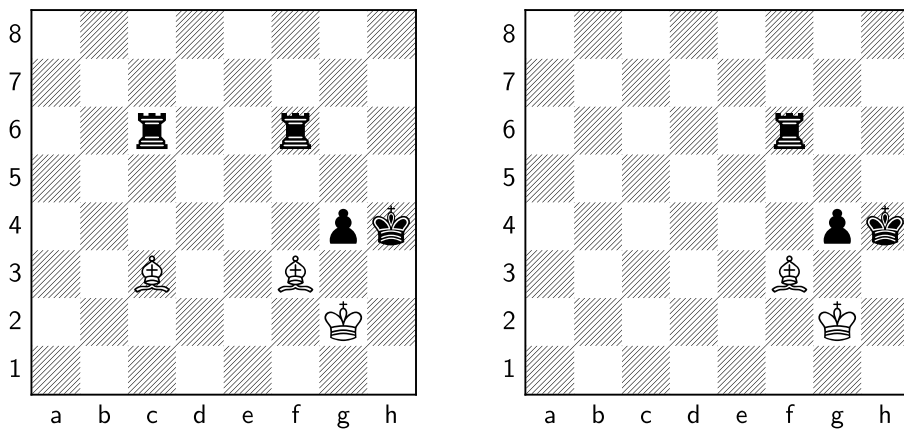


Fig. 7. Positions after move pairs 4 (left) and 5 (right).

White is now free to play the winning move:

6. Bf3-a8#

This example demonstrates the unusual and attacking nature of Reach Chess. White has fewer and lesser pieces than Black and would be in a hopeless position in standard chess, but is able to force the win here. The “reach the far rank” aim supercedes conventional notions of material strength, and Black’s rooks are reduced to simple blockers that ultimately prove unequal to the task.

REFERENCES

- Balaguer, M. (2016). Platonism in metaphysics. In E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab*. Stanford University.
- Browne, C. (2011). *Evolutionary Game Design*. Springer.
- Browne, C. (2019). AI for ancient games. *Künstliche Intelligenz*, 34, 89–93. doi:[10.1007/s13218-019-00600-6](https://doi.org/10.1007/s13218-019-00600-6).
- Browne, C. (2021). Everything’s a ludeme: Well, almost everything. In *Proceedings of the XXIIIrd Board Game Studies Colloquium (BGS (Vol. 2021))*. Paris.
- Browne, C., Soemers, D.J.N.J., Piette, É., Stephenson, M. & Crist, W. (2020). Ludii Language Reference. <https://ludii.games/downloads/LudiiLanguageReference.pdf>.
- Koza, J. (1992). *Genetic Programming*. Massachusetts: MIT Press.
- Pritchard, D.B. (2007). *The Classified Encyclopedia of Chess Variants*. Lynn, UK: Biddles Ltd, King’s.
- Tavener, S. Ai Ai Home Page. <http://mrraow.com/index.php/ai-ai-home/>. (2015–2024).