



## BOOK REVIEWS

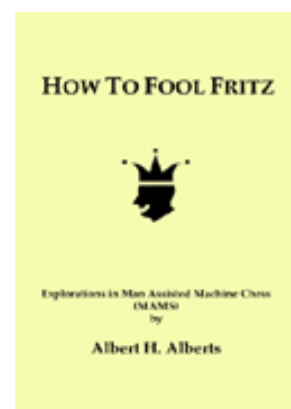


## Fool Me Once...

Steve Lopez

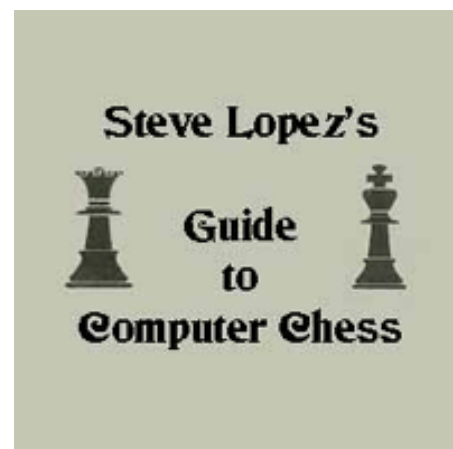
*How to Fool Fritz: Explorations in Man Assisted Machine Chess (MAMS)*, by Albert H. Alberts, 2007 Self-Published, Figurine Algebraic Notation, Paperback, 256pp., \$29.50

Books on how to defeat your chess computer have been on the bookshelves for decades. In the pre-PC era (personal computer, not “political correctness”), chess computers were simply electronic opponents. While some allowed you to set up a position that could then be analyzed, tabletop (sensory board) models couldn’t analyze complete games. These early machines were opponents, pure and simple, not analysts or instructional tools. Nevertheless, a few books offered advice to the frustrated human player on how to defeat the electronic beasts. The earliest example in my library is 1980’s *How to Get the Most from Your Chess Computer* (Julio Kaplan, R.H.M. Press, New York, ISBN 0-89058-046-4), which offered a chapter entitled “How to Beat the Computer.” The advice was fairly rudimentary, but some of it is still useful against many chess engines all these years later: play closed openings, trade material when behind, and try to exploit the “horizon effect” (a machine’s limited capacity to “look ahead,” similar to instances in which a human player captures a pawn only to later discover that it is “poisoned”).



By 1991 a complete book on the subject had been released, namely Raymond Keene and David Levy’s *How to Beat Your Chess Computer* (Henry Holt & Company, New York, ISBN 0-8050-2316-X), which elaborated a bit on Kaplan’s earlier advice. However, the core procedure remained essentially unchanged. Later that same decade, chess software for personal computers offered many advantages over the standalone tabletop models, most significantly the ability to completely analyze the user’s games (instead of just single positions) and point out errors. A chess program began to be viewed by many players as a training and analysis partner instead of just an opponent.

Further volumes on computer chess reflected that trend. The 1998 release *Playing Computer Chess* (Al Lawrence & Lev Alburt, Sterling Publishing Co., New York, ISBN 0-8069-0717-7) offered basic chess advice with illustrative examples drawn from games contested between humans and computers. One chapter, while not exactly a “how to” tutorial, showed how even an average club player could beat an electronic chess opponent. *Chess Software Sourcebook* (Robert J. Pawlak, 2000 Treehaus Books, Montrose, ISBN 0-9673840-01), as well as my own CD *Guide to Computer Chess*, presented advice on using chess computers to help one improve his or her play, while completely omitting



the “how to beat your computer” angle. Speaking for my own publication (released in 2002), I considered the technique to be so well-publicized (primarily on numerous web-sites and message boards) that it didn’t bear repetition on my CD.

The procedure of beating a chess engine isn’t very difficult in *theory*, although the ability to utilize it will tend to vary from player to player. The “fast and dirty” approach that has been successful for many years is to play a closed opening in which the central pawns remain on the board but are locked (i.e. they can’t be exchanged off), wait for the computer to castle, then launch a pawn storm on that side of the board to blow apart the king’s cover, while simultaneously sweeping your heavy pieces to that vicinity (as discussed by Nimzovitch in [Chess Praxis](#)) to deliver the *coup de grâce*.

Of course, sometimes the programmers inadvertently helped. In 1994 a ChessBase customer of my acquaintance discovered a “hole” in *Fritz3*’s opening book. In a particular offbeat opening, the program would always play a specific sequence of moves straight from the opening book (which was provided with the software). The sequence led directly and irrevocably to a blocked center. The Class C player who discovered the “hole” would then follow the “formula” given in the preceding paragraph and defeat *Fritz3* at will. Upon learning of this hole, the programmers altered the opening book for *Fritz4*.

There are also numerous positions in which a chess engine may miscalculate a position because of the “horizon effect” or the effects of pruning (eliminating seemingly unpromising candidate moves early in a search; this is the reason why you don’t typically see a chess computer play queen sacrifices). Computer users post such positions to online discussion groups on a nearly daily basis, asking for input on why the chess program miscalculated a position. These two effects are the usual culprits.

Obviously, it’s no big secret these days that chess engines are, indeed, fallible. So is there really a need for yet another book about the fallibility of chess computers? Albert H. Alberts has produced a new effort in *How to Fool Fritz*, a 256 page opus on chess computer mistakes. However, the book is of questionable merit for a plethora of reasons that we’ll examine below.

Physically the book is of better quality than many self-published chess books. The layout is clear, with generous margins, and the book is bound well. The text itself is marred by numerous typographical errors, especially in the gamescores and variations. When one considers the subject matter of the book (chess software) and realizes that even the most rudimentary chessplaying programs these days have the ability to export games to either PGN format or “plain vanilla” text, there’s really no excuse for typos like “4.Nxc4” (p. 176).

I believe Mr. Alberts is Dutch, which I mention only because of the bearing this might have on the author’s prose style. Grammatically, the book’s text is simply atrocious. The author swings back and forth between run-on sentences and short, clipped sentence fragments. For example, on page 170, we are treated to the following paragraph (which, by the way, is also a single sentence):

*Now a funny question comes up: knowing that a computer will NOT consider these crazy silent moves like for instance Rb8 or Bh6 and no matter how fast and sophisticated engineers and programmers will design chess machines, if four or five irrational human moves are needed to bring advantage to the black configuration and we are by now very familiar with the calculative style of the machine, suppose we play this Shesnikov-variation [sic] from the White side of the board, we should be able to win isn’t it?*

This sentence is fairly typical of Mr. Alberts’ writing style. He will frequently also interject an incomplete sentence fragment into the mix just to spice things up. This makes for some *very* difficult reading.

Every book needs some sort of central theme, a thesis that holds it together. Unfortunately, Mr. Alberts’ thesis is not very clear. He makes several points in the introduction that create a sort of

muddled “theme” should the reader try to piece it together.

*I guess by now machine chess players know that when left alone in the machine-against-machine mode, 99% of the time a game results in a draw. (p. 8). [I dispute the author’s contention. From years of running engine vs. engine games, I’ve noticed typically no more than a 50% draw rate, and the percentage is usually much lower – a far cry from the “99%” claimed by Mr. Alberts.]*

*I will demonstrate that it is fairly easy to beat even the most powerful chess machine... (p. 8)*

*The only time -- ideally -- when we make a ‘human move’ is when taking the risk. At the right moment....the challenge then becomes: at what risk do we revert the game from a ‘bad’ position and what criteria do we obey to when taking the risk. (p. 8-9)*

*... concentrate on developing opening novelties and try to deviate from standard book theory as early as possible in the game. With a constant eye on the risk level. Every bold and overbold move will be investigated. (p. 9)*

*I start to believe that every chess program backed up by arbitrary strong processing devices can be beaten, given enough time and patience. (p. 9)*

*As non-grandmasters we let the machine play the endgame and on top of that we ‘hack’ the openings. Now this is great fun. Trial-and-error is the name of the game. (p. 9)*

*We will not ‘play’ games, we will ‘compose’ them. (p. 9)*

And on pages 8 and 9, Mr. Alberts draws parallels between his ideas and those of Garry Kasparov’s “Advanced Chess,” although Mr. Alberts refers to his concept as MAMS: Man Assisted Machine Schach (Chess).

I will freely confess that I’ve read Mr. Alberts’ introduction many times and I’m still not clear on what he’s trying to say. Is this some sort of computer vs. computer chess with occasional human intervention to correct the chess engine’s errors? Is his technique a form of repeated take-backs in which a human player corrects his own moves with a machine’s help (similar to using “Infinite analysis” mode in the *Fritz* software) until he emerges victorious? Is this really a book about how to beat a chess computer? Is his thesis merely that chess engines make mistakes?

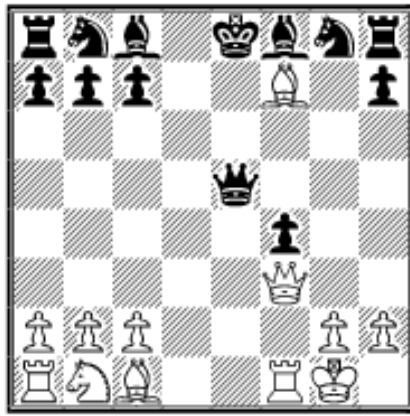
Further reading of *How to Fool Fritz* does little to clarify matters. It’s not simply a matter of the tortured, ungrammatical text; the author seems to jump from point to point, as though he’s trying to prove too many of them, and so ultimately proves none.

Mr. Alberts’ case (whatever it is) certainly isn’t helped by the complete lack of scientific method he exhibits. In his introduction he informs the reader that he is “a research scientist (chemistry-physics),” but I find that very difficult to believe. The sad fact is that I’m often unable to reproduce moves that he attributes to a chess computer’s play.

Let’s examine an example from pages 11-13. The author calls this an “electronic Muzio.” The game begins:

**1.e4 e5 2.f4 exf4 3.Nf3 d6 4.d4 g5 5.Bc4 g4**

Now the author says (in part), “Fritz warns after 0-0 but we will ignore him, hence the symbol M after 0-0(M), a human Man-move.” This is a symbol which Mr. Alberts uses throughout the book.



Here's where we hit several snags. First of all, we don't know which version of *Fritz* Mr. Alberts is using. He hasn't said. Flipping through the book one notices many games he has provided in which he demonstrates his anti-computer technique against *Fritz7* (these inset games are the only cases in which he mentions a program by name and version number – a flaw which we will consider again later), so I think it's safe to assume he's using *Fritz7* as his analysis engine. Unfortunately, this is where we hit another snag. The "(M)" notation after 10.Bf7 indicates that *Fritz7* preferred some other move, presumably inferior, and Mr. Alberts manually forced the move 10.Bf7. Yet, I tested this position using *Fritz7* as an analysis engine,

and the program finds the move 10.Bf7 after a short nine ply search. It continues to suggest 10.Bxf7 Kxf7 11.Bxf4 as its preferred line of play well into double-digit search depths, staying with that sequence until I stopped the engine at fifteen plies. This suggests that Mr. Alberts is utilizing very short searches, allowing the software very little time to ponder and analyze.

The game continues: 11...Kf7 11.Bf4 Qf5. In his notes, Mr. Alberts says that the computer shows an evaluation of -0.82 after 10.Bxf7, but *Fritz7* on my machine shows an evaluation of -1.06 after an eight ply search with varying values at later depths, but never showing -0.82. Later in the paragraph, he provides another variation ending in a 1.39 advantage for White, but my machine shows the position as dead even (0.00) after a fifteen ply search.

So what is Mr. Alberts doing here? Is he running a very slow computer? Or has he tweaked his *Fritz7* engine in some way by using any of the several configurable engine options? It's difficult to tell. The author hasn't provided us with any real information. We don't know what ply depth at which he's stopping the engine. We don't know if he's set any special engine parameters. We don't even know which version number he's using. All we know is that we can't duplicate his results.

We find a clue on page 33 – and this is where the book totally falls apart. Mr. Alberts informs us, "I keep saying 'Fritz' as a collective name for machine chess. I played a whole bunch of programs, my favorite is HIARCS..." Note his words: "*I keep saying 'Fritz' as a collective name for machine chess.*" (emphasis mine). The fact is that **Mr. Alberts can be playing literally any chessplaying program, but doesn't feel the need to tell us which one(s) so that we may verify his techniques.** (As a side note, I went back over the analysis on pages 11-13 using HiarcS, and I still was unable to reproduce the numerical evaluations he reported.)

This may seem harsh, but in my estimation his page 33 "confession" renders the entire book moot. Most of the examples of "weak engine play" that he provides aren't verifiable, because we don't know the engine/version number used. Later in the book (for example on pages 168 and 170) the author indicates that he's using a Nintendo Gameboy chess cartridge for his examples, but this doesn't stop him from referring to them as "Fritz." Mr. Alberts is doing a grave injustice to the programmers of numerous chess engines by referring to them under a generic name. Forgive me for sounding cynical, but using the name of one of the most commercially successful engines in the book's title, and as his "generic" engine name, seems to be merely a calculated (yet shallow) attempt by the author to sell more books.

However, let's look at an example (chosen at random) in which he does give us the name and version number of the software, as well as the size of the transposition tables used to store analyzed positions. On page 176, Mr. Alberts (apparently playing under a nickname) has White against *Fritz7* using 48 MB of hash tables:

1.e4 c5 2.Nf3 Nc6 3.d4 cxd4 4.Nxd4 Nf6 5.Nc3 e5 6.Ndb5 d6 7.Bg5 a6 8.Na3 b5 9.Bxf6 gxf6 10.Nd5 f5 11.Bd3 Be6 12.0-0 Rc8



And here a snag has been found. I researched this game using the *Fritz7* engine using 48 MB of RAM, as shown in the game header. The author has included pondering time for each move, providing times for both himself and the chess engine. I know that there's an opening book loaded because a "0" is displayed for the program's pondering time for each of the game's early moves. The snag occurs on Black's twelfth move (12. Rc8) in which *Fritz7* utilizes over four and a half minutes of pondering time. We can deduce two facts from the time usage. First, the author is using a slow (by 2007 standards) computer. Even on one of my antiquated Pentium III machines using the 48 MB hash tables specified by Mr. Alberts, *Fritz* will settle on that

move in less than a second (a mere six plies). Second, the author has artificially induced that move by either turning off *Fritz*' own opening book (the one that ships on the *Fritz7* disk) or by using a different opening book, possibly a truncated book of his own devising. The normal *Fritz7* opening book contains four candidate moves for Black's twelfth move, none of which is 12...Rc8, and three of which are actually evaluated by *Fritz7* as being *better* for White than the engine's evaluation of 12...Rc8.

Is this what Mr. Alberts is referring to (on page 9) when he mentions "hack[ing] the openings"? It's difficult to know because Mr. Alberts never fully explains his technique (or goal) anywhere in the book. The reader must try to piece it together from various passages throughout the work.

The author comes close to describing both his technique and his goal on pages 239-240. He provides the reader with eight rules to follow when attempting to "fool Fritz." The key rule seems to be #2:

*The trick is to introduce high risk moves in the range of -1 to -3 disadvantage for the human player preferably with the first 10 moves, and create MOBILITY of ALL pieces on his side of the board.*

I presume by "-1 to -3" the author means a one to three pawn disadvantage (rule #5, "Do NOT count the pawns at this stage of the game" seems to support this). This gives us another clue to Mr. Alberts' somewhat murky methodology. He seems to support the idea of playing gambits against chess engines as a means of "fooling" them, since computers often miscalculate such positions by failing to recognize the tradeoff of material for an advantage in development and/or time. If this is indeed his point, there is (again) nothing new here.

In a [column](#) I wrote for ChessBase, I discussed the nature of gambit play: "...the gambiteer's advantage erodes over time, so as the game progresses the engine's evaluation will tend to "normalize" -- that is, as the intangible (in a mathematical sense) advantage gets frittered away, the engine's evaluation becomes more accurate, until it's "spot on": the gambit player's position is a losing one because he's down material and he no longer has a non-material advantage....If you give up two pawns in an opening but see an evaluation of "-1.25," it's a pretty fair bet that you have some level of compensation for the material. It's your job to figure out where that compensation lies and determine how best to exploit it, before that advantage slips away."

Mr. Alberts seems to be saying that you can "fool" a chess engine with properly timed material sacrifices that the engine will miscalculate, giving the human player the upper hand. Yet, on page 9 he says, "We will not 'play' games, we will 'compose' them," and on page 240 he writes, "Compose. Do not play against time. Sit up for days and night if you have to." So, again, it's impossible to determine if we are playing actual games against the computer (as strongly implied by the frequent sidebar games between "Appie" and Fritz7) or if we are manually locating, identifying, and playing around with various gambit-style positions that computers traditionally tend to miscalculate.

The author implies in various places, including the book's introduction, that he's horsing around with positions and allowing himself take-backs and adjustments. From page 9: "Trial-and-error is the name of the game" and on the same page, after the aforementioned remark about "composing" instead of "playing": "Every bold and overbold sacrifice will be investigated."

It's a truism in chess that a beginner can defeat (or at least draw) a grandmaster if the beginner is allowed unlimited take-backs. If one follows this particular interpretation of Mr. Alberts' muddled prose, then one might surmise that the way to "fool" *Fritz* is to allow oneself as many take-backs as are required to "slay the beast."

I freely admit that after numerous readings and perusals of *How to Fool Fritz*, I have absolutely no idea of what Mr. Alberts is about. The text makes for torturous reading. The author never plainly spells out his intent in clear understandable prose. Is the book about chess engines' misevaluation of gambit/sacrifice positions? Is the author advocating occasional human intervention in engine vs. engine games? Is the author presenting a methodology for humans to use in defeating chess computers? All three ideas make appearances, if only briefly, but the author's disorganized, disjointed writing style never fully brings any of these ideas to complete fruition. Mr. Alberts bounces between them like a ping-pong ball, choosing whichever thesis suits the current section's requirements. In the end, it doesn't really matter, because all three of these topics have been presented more clearly in earlier works.

The factor that most hurts *How to Fool Fritz* is the lack of concrete *reproducible* data presented in its pages. Although Mr. Alberts states that he's a research scientist by trade, I find this difficult to believe in view of the slapdash presentation of his game examples. He never discusses the hardware being used, any time controls in force, or whether he's tweaked any configurable engine settings. Most of the time we don't even know which engine he's using, since he's opted to call *any and every* chess engine "Fritz," including a Nintendo chess cartridge. Anybody can *say* they've discovered cold fusion in a jar, but it's quite another thing to *prove* it. I've no doubt that Mr. Alberts has defeated *Fritz* and other chess engines any number of times; if they were impossible to beat, there would have been no market for the many "How to Beat Your Chess Computer" books over the years. But there's not enough hard data in this volume for it to be of any real use to programmers of chess engines, nor enough organized methodology for it to be useful to the average player who owns a chess program as a training/teaching tool. In short, *How to Fool Fritz* seems a better exercise in how to fool buyers of chess books than any real attempt at contributing a useful addition to the existing corpus of computer chess literature.

---

 [TOP OF PAGE](#) [HOME](#) [COLUMNS](#) [LINKS](#) [ARCHIVES](#) [ABOUT THE CHESS CAFE](#)

[\[ChessCafe Home Page\]](#) [\[Book Review\]](#) [\[Columnists\]](#)  
[\[Endgame Study\]](#) [\[The Skittles Room\]](#) [\[Archives\]](#)  
[\[Links\]](#) [\[Online Bookstore\]](#) [\[About ChessCafe.com\]](#) [\[Contact Us\]](#)

© 2007 CyberCafes, LLC. All Rights Reserved.

"**ChessCafe.com**"® is a registered trademark of Russell Enterprises, Inc.