



Chess Tests: The "Never" Concept

(c) Valentin Albillo, 2020



Last update: 27/04/98

See the Notes on Problem Solving

Overview:

The **Never Concept** refers to a situation or action that a human chess player can perfectly see and understand that *it will never happen*, yet a computer chess program is absolutely unaware of this and can easily miscalculate the position completely, thus committing serious and even *fatal errors*, even if searching to extreme depths.

It is easy to give simple examples of this, and the most basic types are already taught to the programs, to avoid gross endgame blunders.

For instance, a King and a Bishop cannot mate a lone King, despite the fact that there is a +3.00 material advantage to the side with the Bishop. A program needs to know this or else it risks entering into an exchange in which it loses its only pawn, perhaps to conserve the bishop !. This would change a possibly winnable K+P vs K endgame to a totally hopeless drawn K+B vs K endgame.

The same can be stated about more unusual material combinations, such as King plus two knights versus king alone, which with perfect play cannot be won either. However, if the side with the king also has a pawn there are many cases in which the two knights *can deliver mate*. Obviously, a program needs to be told of this possibility, else it will easily *capture* the remaining pawn, incrementing its evaluation by +1.00, just to change a possible win for a *dead draw*, once again !

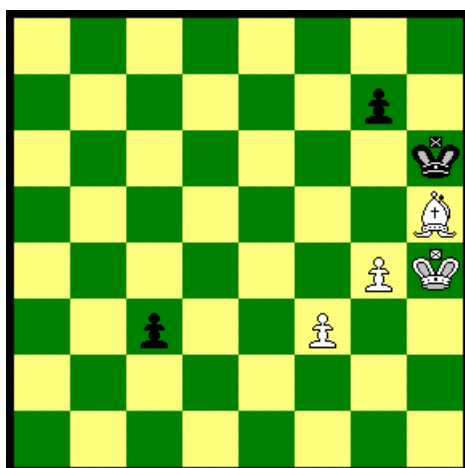
The problem is, there are countless situations in which this **Never Concept** applies, and it seems impossible to program all of them, either as particular cases or with any useful generality.

In this Suite Extension, we will see test positions where one side has an overwhelming material advantage yet *it cannot win* because of some peculiarity of the position. For instance, we will see positions where the weaker side cannot prevent a pawn promotion, yet it can *imprison* the enemy king, so that the newly born queen is useless to win, not being able to checkmate by herself !.

The point is, a human chess player *understands* this and can see without calculation that the situation *can never improve* and thus it's a draw. On the other hand, the program cannot understand any of this, miscalculates the position, and it can either spoil a win trying to reach one of this positions in which it sees a great advantage, only to find it ultimately *unwinnable*, or else spoil a certain draw ruining any essential characteristic of the position that makes it a draw, such as letting the enemy king get out of "prison".

No easy remedy for this. Even if some **Deep Blue 8** could calculate *100 plies ahead*, so that the 50-move rule would show it was an unwinnable position, nothing could be done if such a position were to appear not at the *root* of the search but as a *terminal node*. The program would see a gain or +7.00, say, and would do *anything* to reach that promising, most advantageous position ! *Really sad !*

81.- G. Zajodiakin, 1929



FEN: 8/6p1/7k/7B/6PK/2p2P2/8/8/ w

White to play and draw:

1. g5+ Kh7; 2. Bf7 c2; 3. Kh5 c1=Q; 4. g6+ Kh8; 5. Kg4

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Master 2175	P100	16 Mb	g4-g5+	-5.45	17	00:02:20	can't see the draw
Chess Genius 1.0	P100	320 Kb	g4-g5+	-5.54	15/27	00:01:52	cant' see the draw
PANEK Chess Genius 5.0	PII/266	16 Mb	g4-g5+	-6.15	19/31	00:05:33	can't see the draw
Rebel Decade 1.2	P100	192 Kb	g4-g5+	-4.80	13	00:01:10	can't see the draw
Rebel Decade 2.0	P100	512 Kb	g4-g5+	-4.92	18	00:31:11	can't see the draw
Crafty 12.7	P100	12+5 Mb	g4-g5+	-4.865	13/22	00:01:54	can't see the draw
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	g4-g5+	-4.93	14/23	00:01:54	can't see the draw
KAI Crafty 12.6	Pentium Pro 200 MHz	24+16 Mb	g4-g5+	-4.89	15/23	00:04:57	19.000.000 nodes
KAI Chess Master 5500	Pentium Pro 200 Mhz	?	g4-g5+	-6.25	14/21	00:09:09	can't see the draw
KAI MChess Pro 5.0	Pentium Pro 200 Mhz	10 Mb	g4-g5+	-7.32	12	00:18:03	see notes
BRATON Virtual Chess	Pentium 200 Mhz	64 Mb	g4-g5+	-5.42	20	06:30:00	see notes
BRATON Hiars	Pentium 133 Mhz	8 Mb	g4-g5+	?	18/31	? (overnight)	see notes

Notes:

In this position, we have a fine example of the **Never Concept**. Black's passed pawn is about to promote, and cannot be stopped. A Black Queen *will* appear on the board. However, White can draw by creating a *fortress* to enclose the enemy king.

This Black cannot avoid without allowing White to stop the pawn. The final result is that Black gets his Queen, but his King *cannot* escape from the fortress, and the Queen alone can neither mate the White King, nor separate it from the pawn, so stalemating the King to force the Bishop to move is also impossible. Black can only give check after check, without actually accomplishing anything. *A draw*.

However, most chess programs, if not all, cannot recognize this. They see the Queen on the board and assume Black has a large advantage. They *do not understand* that the Queen is unable to do anything without the King's help, and the King can **NEVER** leave its prison. The problem is, as they do not understand the need of maintaining the King imprisoned, they usually tend to either *move the Bishop*, or separate the King from the pawn, losing the game in both cases.

Chess Master 2175, with a large 16 Mb hash table, finds the correct move, but evaluates it very negatively and thinks it's losing. It does not see the draw.

Chess Genius 1.0, though looking at 15 plies plus 12 extension ones, finds also the correct move, and also thinks it's losing. No understanding of the position, either.

The newer version, **Chess Genius 5.0**, running on much faster hardware and with the immense help of a large hashtable, finds the correct move instantly, looking at 9/21 plies in less than a second, but thinks it's losing by -5.96. When the search reaches 19/31 plies, it still considers it's losing, this time by -6.15. No understanding either. For further comments by **Ed Panek**, see the **Addendum** below.

Rebel Decade 1.2 looks at 13 plies, examines *one million positions*, finds the correct move, and also thinks it's losing.

The newest version, **Rebel Decade 2.0**, searches 5 plies deeper, at 18 plies, taking *30 times longer*, examines *25.694.291 positions*, yet it finds the same move with nearly the same evaluation, -4.92. No draw in sight.

Crafty 12.7 looks at 13 full plies, plus 9 additional plies for a total of 22 on selected branches, finds the correct move and, like the other programs, is fully convinced it is losing.

Crafty 12.6, running on faster hardware and with much greater hash tables, is able to look one ply deeper (14/23) in exactly the same time, and another extra ply (15/23) in triple the time. But though it examines *19.000.000 positions*, it still fails to recognize the draw and thinks it's losing *by the equivalent of a rook*.

Both **Chess Master 5500** and **MChess Pro 5.0** do no better. They search to 14/21 and 12 plies respectively, find the correct move (**MChess Pro** in as little as 0:45), but think they are losing *by even greater amounts* than the other programs. "*Draw ? What draw ?*".

Virtual Chess does one of the greatest efforts to solve this position, running on a fast machine with the largest (64 Mb) hash table and reaching the greatest depth (20 plies) in the longest time (6 hours and a half). Yet it nearly duplicates CM 2175's result, with a heavily negative -5.42 evaluation. The draw goes absolutely unnoticed, despite the fact that the correct move is chosen.

Hiarcs was also tried with this position using a smaller hash table, a mere 8 Mb, but though it reached a depth of 18/31 plies, it couldn't recognize the draw. However, see the **Addendum** below.

So, all the programs tested find the correct move, but simply because it *delays the promotion* a little, not because they see that it draws. It could be argued that it doesn't matter whether they see the draw if they can find the correct moves, but that's not so, as will be explained now.

Without understanding the position, a program is likely to make a fatal mistake. In this position, the fatal mistake is *failing to create the fortress* or, once created, *moving the bishop* or *separating the king from its protecting pawn*. A human player can understand this and will avoid those pitfalls without searching much.

On the other hand, a program, which does not understand any of this and cannot look ahead 100 plies till the 50-move rule saves the day, can and will commit fatal blunders. For instance, playing this position with **Crafty 12.7** results in the following moves:

1. g4-g5+ Kh7; 2. Bg4 Kg6; 3. f4 c2; 4. Be6 c1=Q; 5. f5+ Kh7; 6. g6+ Kh6; 7. Kg3 Qe3+; 8. Kg4

and now Black gives *mate in 9* starting with Qe4+. Here white failed to *create the fortress* and lost the game, despite the first correct move. A similar test with **Chess Genius 1.0** results in White *moving his bishop*, the Black King escaping, and a mate following soon.

Addendum:

Ed Panek tried this position on **Chess Genius 5.0**, and was amazed and amused at the results he got. In his own words:

"... **Genius** finds the draw instantly, but doesn't understand it ... it finds all the right moves at 0 seconds! ... it sees this principal variation:

g4g5, h6h7, h5f7, c3c2, h4h5, c2c1, g5g6+, h7h8, h5g4, c1e3, g4g3, e3g5+

... I will walk **Genius 5** to the point Kg4 and see what it says now that it is sure it's going there ... now in the position where the King is trapped it still says Black is ahead 5.63 at depth 15 ! ... amazing! ... I will now let the computer play itself at each side set to depth 13 plies and see if it can figure out his position ... it plays:

c1e3, g4g3

and **Genius 5** looks like it will lose the pawn and lose :(... Have a good day. Ed."



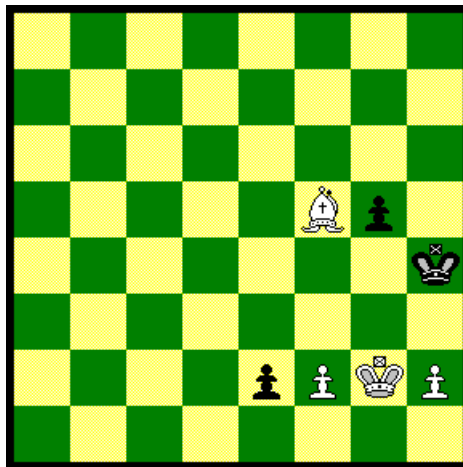
Alan Bratton also tried this position on **Hiarcs**, but to no avail. He said:

"... I also tried it overnight last night on my PC at work with **Hiarcs**. Results: It chooses the proper sequence of moves but doesn't recognize a draw. Line:

1. g5+ Kh7 2. Bf7 c2 3. Kh5 c1Q 4. g6+ Kh8 5. Kg4 Qe3 6. f4 Qd3 7. Kg5 Qh3 8. f5

... (analysis done by loading the **Hiarcs** engine into the **CBLight** program)"

82.- V. Chekhover, 1952



FEN: 8/8/8/5Bp1/7k/8/4pPKP/8/ w

White to play and draw:
1. Bg4 !! e1=Q; 2. h3!

Results

Program	CPU/Mhz	Hash table	Move	Value	Plys/Max	Time	Notes
Chess Genius 1.0	P100	320 Kb	h2-h3	-6.06	16/28	00:57:21	cant' see it
PAÑEK Chess Genius 5.0	PII/266	16 Mb	h2-h3	-6.57	22/32	09:34:22	can't see the draw
NEW Rebel Decade 2.0	P100	512 Kb	h2-h3	-5.12	18	01:46:53	can't see the draw
Crafty 12.7	P100	12+5 Mb	h2-h3	-5.285	24/29	01:43:45	can't see it
Crafty 12.9	P100	6+1 Mb	h2-h3	-5.285	28/30+Hash	26:42:27	can't see it

Notes:

This is another excellent example of the **Never Concept**. White cannot avoid the Black pawn queening, but it can imprison the Black King forever. This is done by the immediate threat to capture the pawn. Black has to queen immediately, as capturing the Bishop would allow White to stop the pawn: 1. ... Kxg4; 2. f3+ Kh4; 3. Kf2.

Once imprisoned, the Black King is out of the game and the Black Queen cannot mate alone, nor can she dislodge the King from its protecting pawns. If this could be done, the King could be stalemated in a corner, thus forcing the Bishop to move, allowing the Black King to escape. But nothing of this can be forced so it's a draw, something any proficient human chess player grasps immediately.

However, current programs do not understand any of this. Their evaluation of the position for White is very negative, as they see the pawn queening. And even worse, they do not find even the correct drawing move this time.

Chess Genius 1.0, at all ply depths up to 16/28 plies, selects *h2-h3* which loses. It takes nearly an hour to search that far, but even so its evaluation for the selected move is -6.06, losing badly.

Chess Genius 5.0, running on a fast machine with a large hashtable, goes very deep, at 22/32 plies (32 plies is its limit), taking many hours, yet it finds the same move and with nearly the same evaluation as CG1.0, -6.57. For extra comments by **Ed Panek** and the *Principal Variation*, see the **Addendum** below.

Rebel Decade 2.0 can't use large hash tables, being limited by design to a maximum of 512 Kb. Yet, after nearly 2 hours it reaches 18-ply depth and selects 1. *h2-h3*, evaluated as -5.12. Though it explored 106.475.789 positions, it couldn't find the correct plan.

Crafty 12.7 also selects the same *h2-h3* losing move at all depths. Letting it go as deep as 24/29 plies takes nearly two hours, yet it evaluates its selected move at a depressing -5.285, also losing.

Just to test the point a little further, I left **Crafty 12.9** look deeply at the position. It reached 28 full plies (plus extensions which finally referred to a hashtable entry) taking nearly 27 hours, yet it did no better than its older incarnation, choosing the exact same move with the exact same value.

These are the first and last lines from **Crafty's 12.9** lengthy analysis, where you can see that depth of search, the always-miraculous substitute for lack of smarts, did *nothing* whatsoever to increase its understanding (or lack thereof) of the position:

depth	time	score	variation
6	0.68	-5.327	h3 e1=Q Bg4 Qe4+ Bf3 Qf5 Bg4 Qe5
...
28	1602:27	-5.285	h3 e1=Q Bg4 Qe4+ Bf3 Qf5 Bg4 Qd5+ Bf3 Qa2 Bg4 Qc4 Bd1 Qd4 Bg4 Qa4 Bf3 Qd7 Bg4 Qb7+ Bf3 Qc8 Bg4 Qc3 Bf3 Qf6 Bg4 Qd6 Be2 Qe6 [HT]

The *Principal Variation* predicted at this maximum depth is:

**1. h3 e1=Q; 2. Bg4 Qe4+; 3. Bf3 Qf5; 4. Bg4 Qd5+; 5. Bf3 Qa2; 6. Bg4 Qc4;
7. Bd1 Qd4; 8. Bg4 Qa4; 9. Bf3 Qd7; 10. Bg4 Qb7+; 11. Bf3 Qc8; 12. Bg4 Qc3;
13. Bf3 Qf6; 14. Bg4 Qd6; 15. Be2 Qe6 and the rest is unavailable (hashtable hit).**

which is *nonsense* from the very start. For instance, after the selected *1. h3*, Black should *not* promote the pawn immediately, as this would allow *2. Bg4*, imprisoning the Black King (not that the program finds it any useful, either), but it should advance the *other* pawn, *g5-g4*, making any imprisonment impossible, and ultimately winning.

Addendum:

Ed Panek, who tried this position on **Chess Genius 5.0**, and let it run for many hours till it reached its limit (32 plies) on the search extensions, included the following comments on the surreal experience:

" ... [the *Principal Variation* predicted at 22/32 plies is:]

h2h3, e2eQ?, f5g4, e1b1 ...

... Now when walked to **bg4!**, **e1eQ**, **h3!!** as proposed correctly by **Valentin Albillo**, **Genius 5** says (depth 16/28) that it will manage to pry off the White Bishop from that square down the wrong diagonal and advance the **g5** pawn ... **nonsense !** ...

... The King may move about 3 squares and the Bishop can move the **g4-c8** diagonal and maintain touch with the **h3** pawn ... position should be static ... for some reason **Genius 5** thinks that its promoted Queen should be able to force some mint move ... maybe against a computer it might, but not against a human player ...

... This is an interesting position because I have myself experienced a position similar to this against computers on **Chess.net** and **FICS** and I offered a draw and the computer I was playing against declined the draw repeatedly. It saw a huge advantage despite the logical flaw ! ... finally I was forced to quit the game and it was adjudicated correctly as a draw !. **Silly computers !**"



(c) Valentin Albillo, 2020