Bug in Multiple Chess Programs: Explained by Kolmogorov Complexity?

Kenneth W. Regan, University at Buffalo (SUNY)

Possible relevance to PRGs, *in-situ* statistical testing, digital simulations, computational depth, life, the Universe, everything, and “10^{500}” other universes…

Slides 7—15 illustrate a demo that was shown “live” using the Fritz 9 GUI when this talk was first given at the Complexity 2007 open session in San Diego.

Chess programs, aka. Engines

- Evaluation function E: Positions $\rightarrow$ R
  - Units are 1/100s of a Pawn, +ve favors White
  - Main distinctive aspect of different programs
    - Champion program Rybka's E is a notorious secret
    - E is "tuned" to master games, and by linear pgmg.
    - E(p) $\in$ {-Inf,0,+Inf} for "tablebased" positions p, such as all p with 5 or fewer pieces left.
- Minimax tree search with $\alpha-\beta$ pruning and iterative deepening of base search depth d
  - Some variants, e.g. "NegaScout".
  - *Extensions* go to depths e>d until *quiescence*.

Zobrist (= Subset-XOR) Hashing

- Zobrist key = a 64-bit key k_f for a feature f
  - f = 1 of 12 different pieces on 1 of 64 squares
  - f = Black, not White, is to move
  - f = White and/or Black can castle; *en-passant*…
- 768+1+4+8 = 781 Z-keys, ~ 50,000 bits, call 'em **B**
- $h_{\mathbf{B}}(p) = (+)\{k_f$: position p has feature f$\}$.
- Estimates 2^{136}---2^{154} legal positions, so h_B has many collisions---even before 2nd-level hashing. Trouble when positions *in same search* collide.
  - [OBDDs can code *meaningful* positions more succinctly, J.T. Kristensen-P.B. Miltersen, 2005-ongoing.]

Indeed, earliest cited case of tabulation hashing is Zobrist, 1970.

Second-Level Hashing

- Chess engines use open-address hashing, often with *no* probing---”Speed is King”!
- Typically 16 bytes per entry, so 512MB hash = 2^25 ~= 32 million entries. h_2: endian or wrap.
- Engines surpass evaluating 1M positions per second---although not all evals are stored, hash table quickly fills!
- On collision, when to replace an eval? Many engines don’t stop to ponder: just overwrite!
- **When/how often does a bad eval propagate to the root of the search tree?** Exp’ly unlikely?
- Shredder 9.0 blunder in 2005 top-level game...

Those 50,000 Z-Key Bits B…

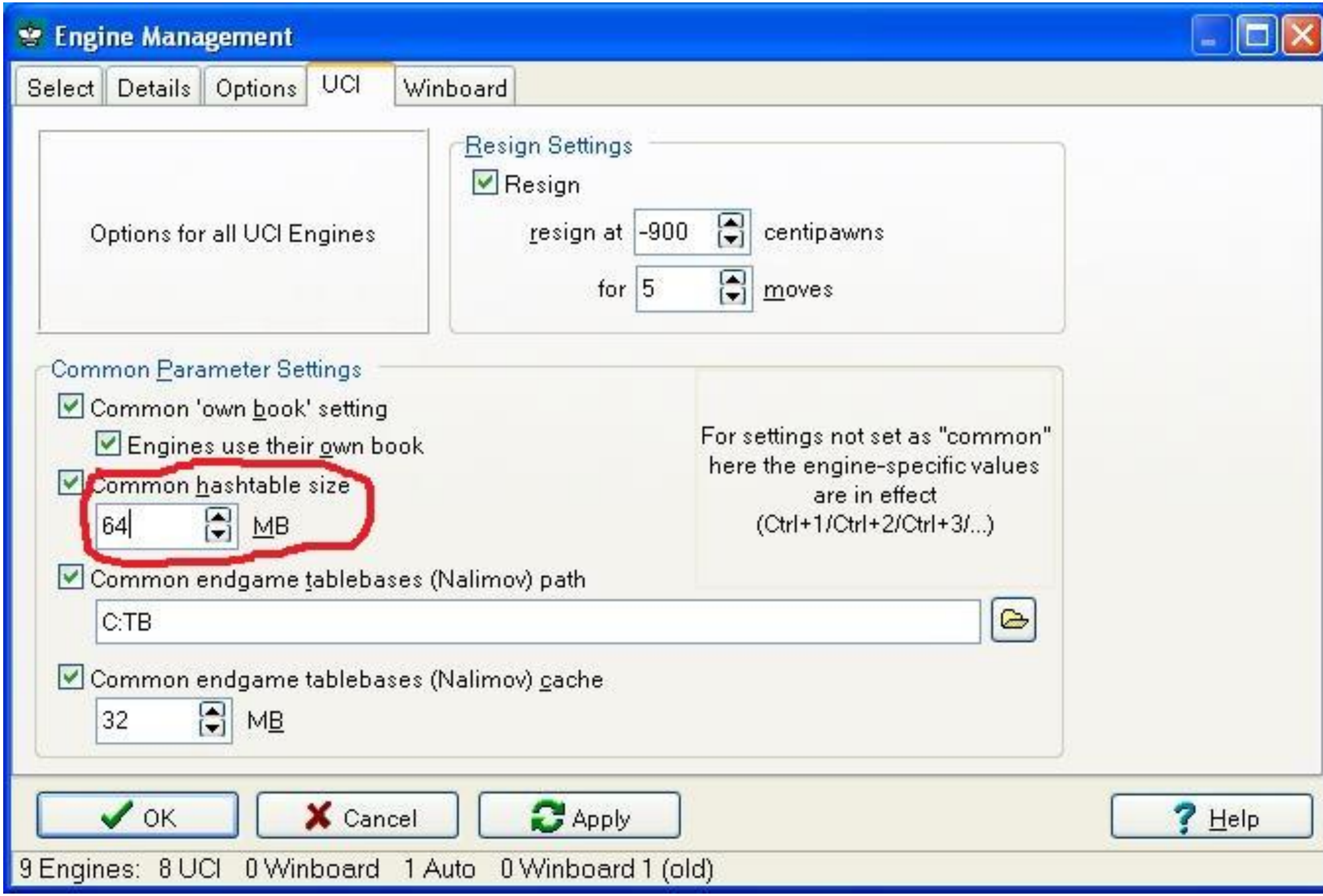
- Should be **random**. (D. Eppstein's notes)
- Are *permanent*---because *opening books* are stored via Z-keys.
- Most (all?) engines generate B by composing PRGs from Mathematica, Knuth ACP, even the Mersenne Twister [MT2002 bugfix].
- ➔ h_B has low description complexity C, K^t, …
- ➔ Colliding p,q have low C(pq) = |s(B),i|
- But maybe not low K^t since one must hunt to find the i-th colliding pair? (→ Computational Depth)
- Can save |i| too if p,q extremize some other predicate ---such as malign effect on the search!

Possible Effects of Low C(B)?

- Not that hash collisions are more frequent: PRGs used for B pass all linear stat. tests.
- *Nonlinear* effects on interaction with *depth-first search*, in main body of engine.
- Perhaps low C(pq) causes colliding p,q to arise more often in search branches…
- …and to arise at critical points?
- Possible *malignness bias* of low-KC strings? Vague, but quite general: Li-Vitanyi 1991++, Jagota-Regan 1991+, Miltersen 1991+.
- A hash collision caused the Shredder 9.0 program to blunder a Bishop and lose a tournament game in 2005. I reproduced it--only at low (2MB) table size.

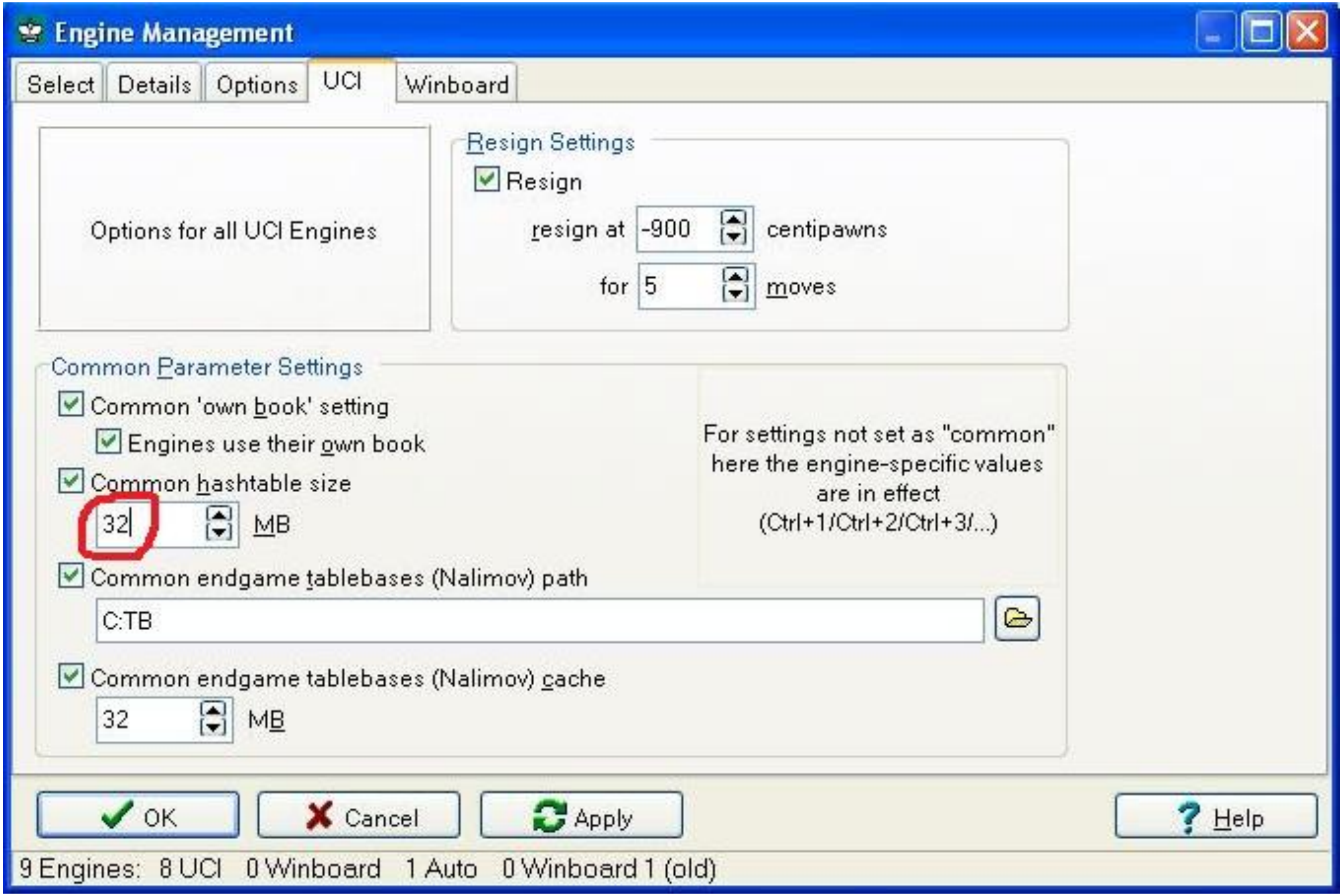
A Wider Reproducible Example

- Download Toga II 1.4 beta 5c 1-cpu.
- Download and install Arena 3.0 chess GUI.
- Install Toga II as a UCI Engine.
- Open file TK74a.pgn, select first item---a position from analysis of a game between Veselin Topalov and Vladimir Kramnik.
- Position is objectively drawn but tricky--many programs are deceived like what follows..
- Click on 74…Rb7+, so it's White's move 75.
- Click Engines→Manage→UCI tab, and do:

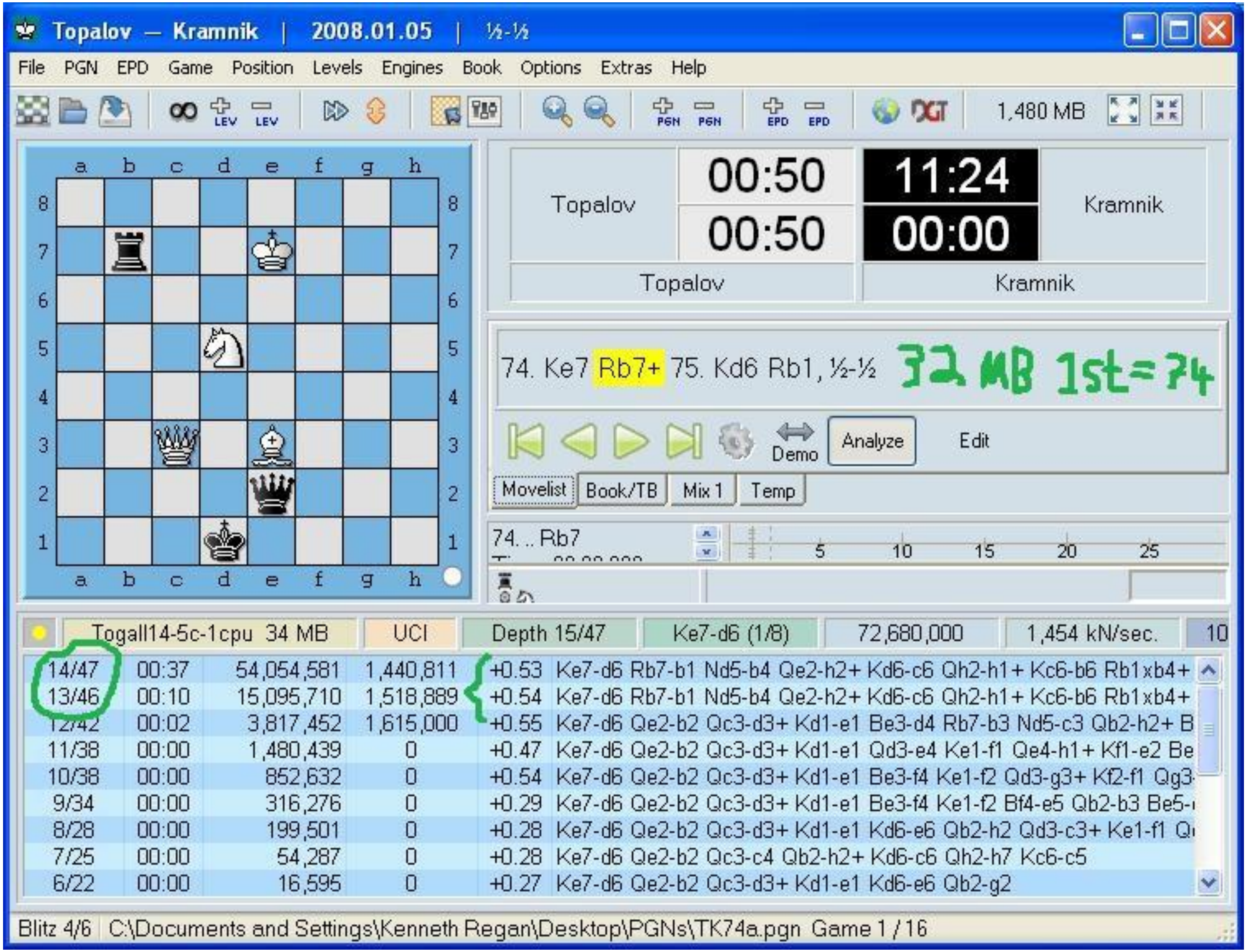


Set hash size to 64MB, click OK. Then hit the “Analyze” button.

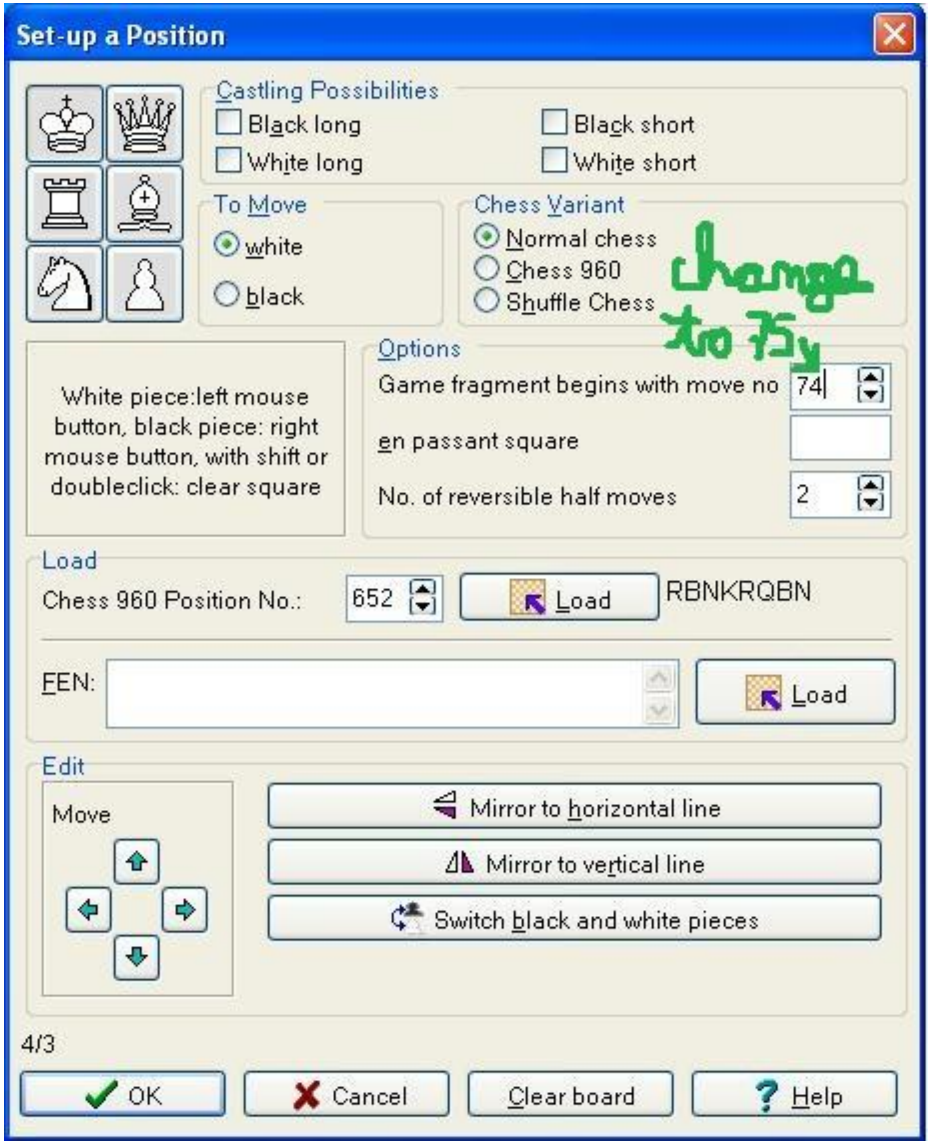
Exit program, reload, select same game, set hash to 32MB



Then hit OK, click Analyze, and see different results…

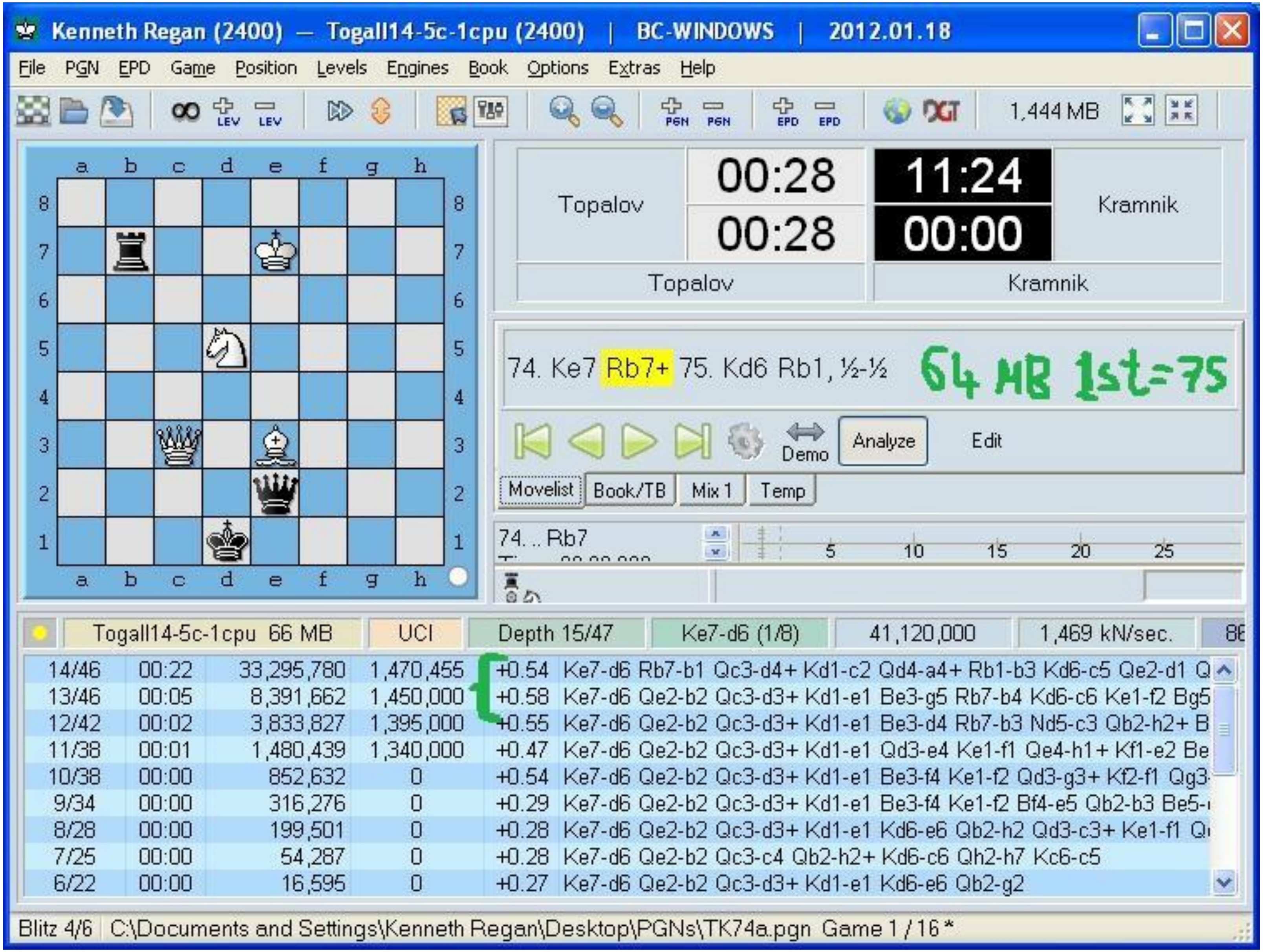


No anomaly! But now select Position→Setup and change the origin move…



As the chess engine's programmer Thomas Gaksch explained to me, even though move 74 still shows in the game notation, this tells the engine to begin the "Fifty Move Rule Count" right here, without the prior move 74. This makes a minuscule difference in the evaluation function---but it is enough to switch around the effect!


Kenneth Regan (2400) — Togall14-5c-1cpu (2400) | BC-WINDOWS | 2012.01.18
File PGN EPD Game Position Levels Engines Book Options Extras Help
1,472 MB
Togall14-5c-1cpu
00:28
00:28
04:00
00:00
Togall14-5c-1cpu
Togall14-5c-1cpu
Togall14-5c-1cpu
74. Ke7 Rb7+ 75. Kd6 Rb1, ½-½
32 MB 1st=75
Demo
Analyze
Edit
Movelist
Book/TB
Mix 1
Temp
Kenneth Regan -
Togall14-5c-1cpu 34 MB
UCI
Depth 15/47
Ke7-d6 (1/8)
41,520,000
1,483 kN/sec.
14/46 00:18 27,951,548 1,512,778 +0.55 Ke7-d6 Rb7-b1 Nd5-b4 Qe2-h2+ Kd6-c6 Qh2-g2+ Kc6-c5 Rb1xb4
13/46 00:08 12,719,630 1,515,000 +6.84 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-g5 Rb7-b4 Qd3-g3+ Ke1-e2
12/42 00:02 3,817,452 1,450,000 +0.55 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-d4 Rb7-b3 Nd5-c3 Qb2-h2+
11/38 00:01 1,480,439 1,360,000 +0.47 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Qd3-e4 Ke1-f1 Qe4-h1+ Kf1-e2
10/38 00:00 852,632 0 +0.54 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-f4 Ke1-f2 Qd3-g3+ Kf2-f1
9/34 00:00 316,276 0 +0.29 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-f4 Ke1-f2 Bf4-e5 Qb2-b3
8/28 00:00 199,501 0 +0.28 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Kd6-e6 Qb2-h2 Qd3-c3+ Ke1-f1
7/25 00:00 54,287 0 +0.28 Ke7-d6 Qe2-b2 Qc3-c4 Qb2-h2+ Kd6-c6 Qh2-h7 Kc6-c5
6/22 00:00 16,595 0 +0.27 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Kd6-e6 Qb2-g2
Blitz 4/6 C:\Documents and Settings\Kenneth Regan\Desktop\PGNs\TK74a.pgn *

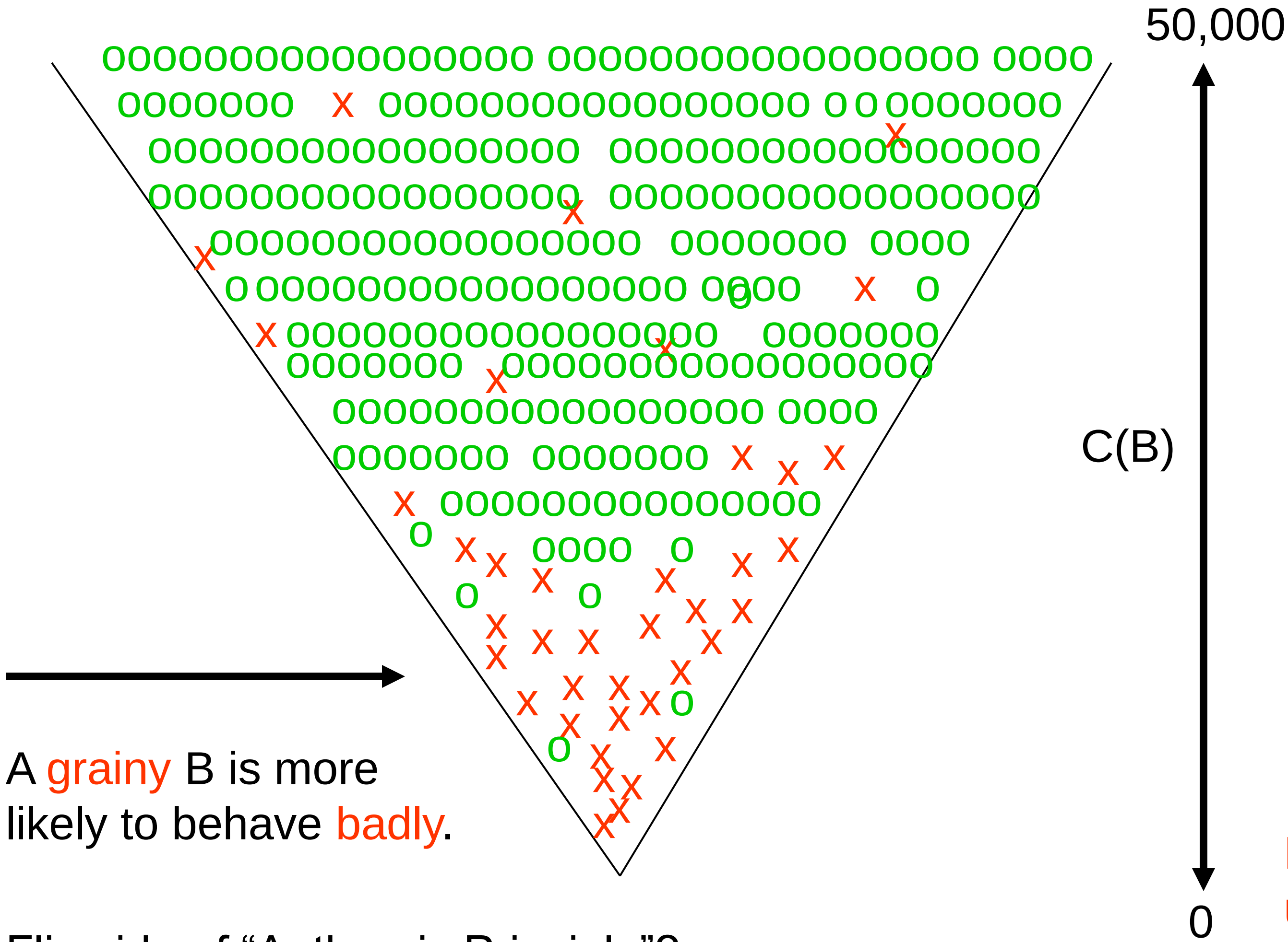

Kenneth Regan (2400) — Togall14-5c-1cpu (2400) | BC-WINDOWS | 2012.01.18
File PGN EPD Game Position Levels Engines Book Options Extras Help
1,444 MB
Topalov
00:28
00:28
11:24
00:00
Kramnik
Topalov
Kramnik
74. Ke7 Rb7+ 75. Kd6 Rb1, ½-½
64 MB 1st=75
Demo
Analyze
Edit
Movelist
Book/TB
Mix 1
Temp
74. .. Rb7
Togall14-5c-1cpu 66 MB
UCI
Depth 15/47
Ke7-d6 (1/8)
41,120,000
1,469 kN/sec.
14/46 00:22 33,295,780 1,470,455 +0.54 Ke7-d6 Rb7-b1 Qc3-d4+ Kd1-c2 Qd4-a4+ Rb1-b3 Kd6-c5 Qe2-d1
13/46 00:05 8,391,662 1,450,000 +0.58 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-g5 Rb7-b4 Kd6-c6 Ke1-f2 Bg5
12/42 00:02 3,833,827 1,395,000 +0.55 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-d4 Rb7-b3 Nd5-c3 Qb2-h2+
11/38 00:01 1,480,439 1,340,000 +0.47 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Qd3-e4 Ke1-f1 Qe4-h1+ Kf1-e2
10/38 00:00 852,632 0 +0.54 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-f4 Ke1-f2 Qd3-g3+ Kf2-f1
9/34 00:00 316,276 0 +0.29 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Be3-f4 Ke1-f2 Bf4-e5 Qb2-b3
8/28 00:00 199,501 0 +0.28 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Kd6-e6 Qb2-h2 Qd3-c3+ Ke1-f1
7/25 00:00 54,287 0 +0.28 Ke7-d6 Qe2-b2 Qc3-c4 Qb2-h2+ Kd6-c6 Qh2-h7 Kc6-c5
6/22 00:00 16,595 0 +0.27 Ke7-d6 Qe2-b2 Qc3-d3+ Kd1-e1 Kd6-e6 Qb2-g2
Blitz 4/6 C:\Documents and Settings\Kenneth Regan\Desktop\PGNs\TK74a.pgn Game 1 / 16 *

What causes it? What does it mean?

- The anomalous evaluation has been isolated to both the hash table size and the 50-move rule component.
- The latter is a “Digital Butterfly Effect.”
- The former operates in both cases.
- It can also be varied by choosing one of the other 15 rotations/reflections, which use different hash keys.
- Clearly an effect of hash collisions propagating to the root of the search tree.
- Queries on how hash key bits used in all common Fruit/Toga versions were generated not yet answered.
- Full investigation of this phenomenon will require much larger-scale testing and modifying source code.
- But for now we can speculate…

General Hypothesis

A random B is near-certain to behave optimally well.



Among choices of "decent" PRGs g of low C(g) used to generate B, it may be highly likely that Fritz, Fruit, Zap!, Naum etc. use "bad" ones for whatever performance metric is relevant.

But, Glaurung 1.2.1 uses MT2002!

A grainy B is more likely to behave badly.

Flip side of "Anthropic Principle"?

Freakier Explanations?

- Low $C(h_B)$ causes errors to "synchronize"-? If positions p',q' follow "shortly" from colliding p,q, then C(p'q') is also low, perhaps making them more likely to collide?
- "Extended Occam Hypothesis"?: Data d with low C(d) [or low $K^t(d)$] arise more often. (Cf. J. Schmidhuber, "Speed Prior") →LHC binary data?
- Are parameter settings that extremize simple functions (such as Smolin's black-hole formation?) more likely? Solomonoff-Levin not Lebesgue $dist^n$. for "Landscape"?
- If C(Y|X) << C(X), C(Y), does Y correlate with X, even though X may not "cause" Y? "An acausal connecting principle"…

Stat. Test for low C(g)?

- Use PRG g to generate Z-keys B for one of the mentioned chess engines E (or any engine, or any nonlinear digital system?)
- Run E on test-suite of positions designed to maximize effects of hash collisions, or on billions(!) of random positions/configurations...
- Provided that random B ➔ best possible behavior (not yet tested), misbehavior ➔ low C(g).
- Low C(g) may cause *unwanted emergence* in digital simulations that require a “smooth random background.” Molecular simulations, more?