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 $\rightarrow \mathbb{R}$
  - 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.
    - $E(p) \in \{-\infty, 0, +\infty\}$ 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, \( \sim 50,000 \) bits, call ‘em \( \mathbf{B} \)

• \( h_B(p) = (+)\{k_f : \text{position } p \text{ has feature } f\} \).

• Estimates \( 2^{136} - 2^{154} \) legal positions, so \( h_B \) has
  many collisions---even before 2\(^{nd}\)-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 = \(2^{25} \approx 32\) 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 \( C, K_t, \ldots \)
- 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?
• 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!
74. Ke7 Rb7+ 75. Kd6 Rb1, \( \frac{1}{2} - \frac{1}{2} \)
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.

A grainy \( B \) is more likely to behave badly.

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.

Flip side of “Anthropic Principle”? But, Glaurung 1.2.1 uses MT2002!
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 $\rightarrow$ best possible behavior (not yet tested), misbehavior $\rightarrow$ low C(g).
• Low C(g) may cause unwanted emergence in digital simulations that require a “smooth random background.” Molecular simulations, more?