The myth Rajlich copied Fruit

Revision 1.1

Contents

September 11, 2015

The ICGA Rybka–Fruit case (where everybody lost) in a nutshell presented with as less as possible technical issues.

Ed Schröder

The myth Rajlich copied Fruit

THE MANTRA

Zach Wegner – I came to the conclusion, after seeing what I saw, that Rybka started its life as Fruit.

Zach Wegner – It’s very clear to me that the reason Rybka 1.0 was strong was that it took Fruit 2.1, tuned the parameters, made some other inconsequential changes, and sped it up quite a bit. Why anyone would want to consider Rybka an original engine in light of the pages I posted is beyond me.

Zach Wegner – it’s just so obvious to me that Vas took Fruit as a base and rewrote things on top of it, presumably until he felt it was “clean”.

Robert Hyatt – He’d have fared much better in “the court of public opinion” had he simply said “I started with the fruit source, but have modified it heavily and today little if any of that code remains.” We didn’t get that kind of honesty however. We got “silence”.

Robert Hyatt – Vas clearly copied fruit to create Rybka 1. Absolutely no doubt. And yet he has repeatedly said he copied _no_ code whatsoever, just ideas. That’s false.

Robert Hyatt – I have no idea what that means. Copying an open source program, closing it and calling it your own is not exactly ethical is it? you get the point…

Mark Watkins - One example of copying seems to be in how Rybka parses the "position" string.

Mark Watkins - I think the case for "copyright infringement" (or plagiarism) of the evaluation function as a whole is quite weighty, particularly when combined with the various other Fruit 2.1 bits that appear here-and-there in Rybka 1.0 Beta.

For years (2008-2011) statements like these were thrown into the public repeated enough to become a reality for many. And in the end the leading Rybka accusers (Watkins, Wegner, Hyatt) couldn’t proof it. The ICGA verdict doesn’t mention a copyright nor GPL breach. At least that part they got right.

And the evidence for that is indeed extremely poor, you cannot use a couple of code snippets to conclude *Rybka started its life as Fruit.* [Wegner] [Hyatt]

We will address these code snippets in question in Chapter 7 (Technical Stuff). For the moment we like to focus on non-programmer issues.

Mixed Messages

Rybka investigator **Mark Watkins** – The Rybka 1.0 Beta executable contains no literally copied evaluation code from Fruit 2.1.

Rybka investigator **Zach Wegner** (in contradiction to his above statements when after the ICGA verdict his document became under pressure) – Coming back to Rybka/Fruit, yes, **if you look at each example in detail**, you can't say there is much hard evidence of direct code copying. As I said to Vas, I'm only completely certain that three characters were copied ("0.0″).

Indeed, when we go through each **evaluation in detail** the accusation "*Rybka started its life as Fruit"* falls into pieces. Also it's odd they contradict themselves, see above statements.

In short: Every major Rybka evaluation ingredient **evaluates different** than Fruit at abstraction level: mobility, king safety, passed pawns, double pawns, backward pawns, Rybka is missing Fruit's late endgame knowledge, Rybka has a material table different from Fruit. Rybka does not contain Fruit's quad function, Rybka's trapped bishop evaluation is different, rook evaluation is different, bishop pair evaluation different and **most important,** Fruit evaluates in stages in a unique way whereas Rybka adds directly to its score (as every other program on the planet does).

We found 36 indisputable differences, some of which are very fundamental.

In this document I like to use statements of other chess programmers who have studied the case. For instance, about the accusation Rybka basically is a bit-board rewrite of mail-box Fruit, user MEC at Rybka forum stated:

User MEC - NO reasonable programmer would ever want to 'copy' the source code of some other program - if he knew, that he'd need to rewrite 90% of the source code in order to produce the result he desires.

If you look up user **MEC** you will see he joined 19 days after the verdict, posted 45 times, left 4 months after. Country Austria. He is a well informed chess programmer with a strong anti ICGA stance on the matter.

How much is too much?

With the ICGA verdict in mind (no copyright nor GPL breach) the ICGA basically has convicted a programmer for taking **too much** ideas of an open source program as confirmed by Mark Lefler and Robert Hyatt of the ICGA Secretariat.

Schröder – Perhaps this split and divided community can settle on: He took too much.

Lefler – Ed, I think that is the best summary of this whole thing. Vasik took too much in the eyes of the panel.

And:

Schröder – In the end all they have proven is that Vas spoke the truth when he on multiple occasions stated: **– I took many things from Fruit**.

Hyatt – We don't disagree with the "I took many things from Fruit." That's obvious. But it could be reworded "I took TOO MUCH from Fruit".

And so the real questions are:

1. What is too much? And where in rule #2 is that described, defined?
2. If it's not defined, then who decides what is too much?
3. **How many ideas is a programmer allowed to take from an open source?**
4. How could a programmer in 2005 know the limits of what is allowed taking ideas under the then valid rule #2?

And what's the own responsibility of the ICGA in this matter? Doesn't a programmer have the right to know where the **exact** line is drawn into the sand?

Apparently rule #2 is not so clear where the line is drawn, the LION++ case of 2006 convincingly has proven that. The programmers of the LION++ team consulted a lawyer before entering the WCCC and were disqualified nevertheless.

A rule change followed because of that unfortunate incident but still no guidance for programmers.

About originality

Rule #2 – **Each program must be the original work of the entering developers.**

The question arises, what is originality? The answer will depend from person to person and yet the word "original" is the key word in their only rule in order to participate. Multiple times I have asked the question, define original and the answer did not come.

There was already enough originality in Rybka 1.0 (The MIT, now in use by many, the usage of 2 scores, one in EVAL (3200/3399), one in SEARCH (100), new search techniques).

Originality is also about ELO. Rajlich revolutionized computer chess, in a couple of years he was able to gain 400 elo points, unprecedented in the history of computer chess, moving from approx. 2700 to 3100 ELO.

To use an analogy, consider the scientific community stripping Albert Einstein from all his honors, esteem, rewards etc. because his theory of relativity and $E = mc^2$ is not ORIGINAL because Einstein took "too much" from the work of Isaac Newton.

Or better let him speak for himself.

https://www.youtube.com/watch?v=KqwahEtUYNo

One last thought

With the ICGA verdict in mind (and how rule #2 possibly can be interpreted with serious life changing consequences) a programmer is left with the dilemma:

- Shall I study open sources and gather knowledge? Tricky, the risk is too high I am influenced too much by it.

Or…

- Shall study open sources just to make sure I don't have anything similar to them in my own code, just in case the ICGA is tempted to peek into my binary?

As for a personal note – I have wandered for 15+ years in ICGA tournaments and it never crossed my mind that rule #2 could be used as an obstacle for progress.

User MEC - inquisition (and especially the life-time ban) have nothing to do with Vas breaking their Rule #2 nor any other Rule at all, because such a Fruit-specific Rule didn't ever exist. If it did exist, it wasn't written anywhere, and thus nobody (except VR's haters) could have known about it. This bunch of inquistors spent the years 2005-2011 with nothing else but 'inventing' an additional ICGA-rule, which didn't exist in 2005. It's safe to say it's the so called Fruit-Rule. Only now we suddenly come to know, why Vas broke the ICGA-Rule #2 six years ago: Because he couldn't foretell that 2011 Hyatt & his fellows dictated, which Re-Use of Fruit's-Ideas was allowed in 2005, and which Re-Use was NOT allowed. Too bad they didn't tell Vas about their Top-Secret Fruit-Originality Rule in 2005.

ICGA Responsibility

Was Vas inspired by Fruit’s eval? Yes, he honestly indicated to that by his statement in December 2005, a cheater likely never would, we are not aware of any example. Did he modeled Rybka’s eval to Fruit’s? Maybe to likely. Did he verbatim copy Fruit’s eval? No. Is it possible after studying an open source (forwards and backwards using Vas’ own words of December 2005) to erase the influence from your mind and to what extend? Difficult.

Let’s say Vas modeled Rybka’s eval to Fruit’s as a hypothesis and by doing so he overstepped the boundaries of rule #2 **then where does rule #2 states that it is forbidden?**

Did the ICGA ever bothered to modernize and adapt to the changing world of internet and strong open sources?

Let’s be reasonable, the crime (if there was one) doesn’t fit the punishment.

Certainly the ICGA is part of the arisen problem itself (see previous chapter) by the lack of clear rules and until this day the ICGA refuses to take responsibility for their own failures in the Rybka / Fruit controversy.

User MEC - I guess now I've clearly DEBUNKED the malicious intentions of the ICGA-inquisitors, all they ever wanted is the total destruction of VR as a person! Those ICGA-inquisitors are nothing else but abyssally vitriolic buggers who have nothing else to do in their spare time but finding evidence which they can use in order to ruin VR. Spending 6 years for finding some dubious evidence testifies how obsessed they were with damaging VR's reputation & VR as a person. Even now they still can't get enough and keep on spreading their lies all over the internet until VR's lifework Rybka is completely destroyed by the FSF.

The evidence about “TOO MUCH”

So which ideas (and how many ideas) did Rajlich really took from Fruit? We take Watkins EVAL-COMP (another document used to incriminate Rybka) as our guide.

While we in generally agree (except a number of exceptions) with the descriptions of each EVAL ingredient given we reject the comparison with 7 other engines out of first hand because it is off-topic. The Rybka/Fruit controversy is not about other engines, it’s about:

Zach Wegner – Simply put, Rybka’s evaluation is virtually identical to Fruit’s.

Mark Watkins – I think the case for “copyright infringement” (or plagiarism) of the evaluation function as a whole is quite weighty, particularly when combined with the various other Fruit 2.1 bits that appear here-and-there in Rybka 1.0 Beta.

And so instead:

1. We will only measure the similarity between Rybka and Fruit.
2. Because of the above strong accusation our measurement will be different than Watkins because every eval ingredient Watkins hasn’t rewarded with 1.0 means that Watkins **BY HIS OWN ADMISSION** states that the Rybka code ! = Fruit and/or evaluates different than Fruit, thus, isn’t **SIMILAR**.
3. Therefore our measurement will be about 100% similarity only (either 0.0 or 1.0) and not something in between. In the light of the **2 above strong accusations** we consider this the only fair approach.

As such we get another insight what Rajlich supposedly took from Fruit using the data from Watkins as demonstrated in the below Excel sheet:

EVAL_COMP

From Watkins data we learn that from the 47 EVAL ingredients he investigated that:

Rybka 1.0 vs Fruit 2.1 similarity = 13
Rybka 2.32 vs Fruit 2.1 similarity = 7

Of which 4 are definitely false (see chapter 9), so:

Rybka 1.0 vs Fruit 2.1 similarity = 9
Rybka 2.32 vs Fruit 2.1 similarity of only 4

Rybka 2.32 being the version that played in Amsterdam 2007 and became Rybka’s first ICGA world champion title. Rybka 1.0 is the version that never participated in an ICGA tournament.

You can make up your own mind on the 4 issues we disagree with Watkins (see chapter 9) and whether the 2 orange listed eval ingredients (which are so common) deserve a place in Watkins document looking at the goal of what the document is trying to explain, similarity. Not counting them would drop Rybka 1.0 similarity to 7 and Rybka 2.32 to 3.

Technical stuff

The Fruitification of Rybka

“**If the facts don’t fit the theory, change the facts**.”
Albert Einstein

In science you are supposed to present the data as it is, transparently.
Miguel A. Ballicora (author of Gaviota)

“**In retrospect, and from my point of view, that [ICGA] Panel was just setup to lend credibility to the desired outcome, and nothing else**.”
Marcel van Kervinck (Panel member and author of Rookie)

We have noticed the tendency by the Rybka accusers to change the facts to fit their predefined “guilty” theory in such a way Rajlich is always guilty no matter what the code tells, a typical -/- always lose scenario. We like to present a couple of examples. We have called the phenomenon, **The fruitification of Rybka**.

We like to present 4 of such examples, example-4 being the worst of all to our taste but your mileage may vary.

Example-1

In Zach's document the section Material states:
The material tables in Rybka were one of the more interesting features introduced. Their implementation was a new way to evaluate material imbalances. The indexing and evaluations in the table seem to be unique, **but there are some very interesting similarities in the information stored in the table with Fruit.**

Typical case of "fruitification". While Rybka's material table (over 1 Mb and stored in Rybka's source code) is a brand new idea in computer chess and admitted by Zach he could not resist the temptation to link it with FRUIT anyway. And so his whole document breaths one thing: GUILTY. How unscientific.

First of all Fruit has NOT a material table such as Rybka, Fruit has a hash-table that is maintained during search whereas Rybka has a fixed large pre-computed material imbalance table inside the executable with a size of over 1MB. Rybka has almost all things different compared to Fruit but in the Wegner's document it is almost identical.

Also note the various assumptions made.
Zach Wegner: (1) – The structure of the material table (at the source code level) **isn't certain**. It **seems likely**, based on the disassembly, that the data type **is something like this**: (2) – The use of unsigned char and bool for phase and lazy eval **are quite likely** (3) – The flags fields **are unclear**, though (4) – This compression of two bytes to one byte was **most likely** done so that….[page 26]

Further refutation.

Example-2

Fruitification, in action, things get more serious now.

ROOK evaluation as mentioned in Zach's document. Check out the screenshot of Zach's document and notice what is wrong with that. A lot. It was not noticed by the ICGA Panel but CSTAL author Chris Whittington was keen enough to see that a compiler would merge "RookOpenFileOpening – RookSemiOpenFileOpening" into one value since both are declared as a CONSTANT.

So where are these 2 constants (1035 and 428) coming from? They are not in the RYBKA executable, only the values 971 and 172 are.

Zach document:

```
const int RookSemiOpenFileOpening = 64;
const int RookSemiOpenFileEndgame = 256;
const int RookOpenFileOpening = 1035;
const int RookOpenFileEndgame = 428;

opening += RookOpenFileOpening - RookSemiOpenFileOpening;
endgame += RookOpenFileEndgame - RookSemiOpenFileEndgame;
```

RYBKA original ASM code

```
401b50: 48 85 05 39 ca 26 00      test QWORD PTR [rip+0x26ca39],rax
401b57: 75 3d                     jne 0x401b96
401b59: 83 c6 40                  add esi,0x40         // 64
401b5c: 81 c7 00 01 00 00         add edi,0x100        // 256
401b62: 4c 85 f8                   test rax,r15
401b65: 75 0c                     jne 0x401b73
401b67: 81 c6 cb 03 00 00         add esi,0x3cb        // 971 (1035-64)
401b6d: 81 c7 ac 00 00 00         add edi,0xac         // 172 (428-256)
```

And so Zach in order to prove FRUIT code (semantics) ASSUMED Constant Folding taking place as part of the Optimizing Process of the compiler.

When confronted with the evidence the main accusers Robert Hyatt, Mark Lefler, Wylie Garwin, Zach Wegner and Mark Watkins responded as follows:

1. Wylie Garwin: promised an answer but never gave one.

2. Robert Hyatt: This is simply another example of "highly deceptive and dishonest statements."

3. Mark Lefler: offering only assumptions.

4. Zach Wegner: no reply.

5. Mark Watkins: *Zach* is the more relevant person to ask about his document.

All in denial which is far more worse than the fruitification issue itself.

These were the leading people responsible for the ban of Rajlich.

In science, unfolding a constant “A” into “B – C” just because you **assume** it is justified, **without any explanation**, is literally, academic dishonesty. In science you are supposed to present the data as it is, transparently.

The fact that the 5 accusers do not realize how bad this is we consider problematic.

Example-3

One more of the many.

Rybka quite differently evaluates than Fruit, using Bishop Mobility as an example:

```
op[me] += mob * BishopMobOpening;
eg[me] += mob * BishopMobEndgame;
```

And then later "op" and "eg" are merged with the 2 real scores.

```
*opening += ((op[White] - op[Black]) * PieceActivityWeight) / 256;
*endgame += ((eg[White] - eg[Black]) * PieceActivityWeight) / 256;
```

WHILE Rybka directly adds to score (opening and endgame).

```
401b67: 81 c6 cb 03 00 00 add esi,0x3cb    // 971 (1035 - 64)
401b6d: 81 c7 ac 00 00 00 add edi,0xac     // 172 (428 - 256)
```

Fruit evaluates different than Rybka. Fruit (unusually) evaluates in 2 steps, Rybka (normally) directly adding to the score.

When we pointed to this (obvious) and quite fundamental difference the ASSUMPTION trump card was played once again notable by the 2 leading people from the ICGA Secretariat.

Mark Lefler: My understanding was this was on purpose. Fabien wrote the code so it would work from either side, so he just looped over the to sides and correct for black by subtracting. **A classic speedup trick** is to unroll the loop, and this is what was done in Rybka. Save a little time since the two score do not need to be combined.

And Robert Hyatt in response to Mark Lefler:

Robert Hyatt: **Very possible**. The primary point, however, is that we have a very small group (2 actually) that want to claim dishonesty, misrepresentation, and a host of other things, since they have, apparently, given up on discrediting the evidence itself. So now we are entering state III of this process (the first was denial, the second was the evidence is flawed) where the people involved are simply trying to deceive everyone by pointing out flaws that do not exist and which do not make any sense whatsoever.

Another case of reversal of the burden of proof followed by the classic
"*if you can't refute the argument attack the messenger*".

Example-4

Objectivity test

Recently we decided to test the objectivity of the accusers once more to see if something had changed in their line of thinking during the years. But sadly they gave the **exact answer** we expected.

We took something from Mark Watkins EVAL_COMP document what was plainly wrong. It states the Fruit and Rybka evaluation of a double pawn are 100% identical at abstraction level.

We pointed out this is not true providing the code of both programs.

```
FRUIT                         RYBKA
if (doubled) {                if (doubled) endgame -= 158
opening[me] -= 10;
endgame[me] -= 20; }
```

There is simply NO midgame code in Rybka.

And as predicted both Robert Hyatt and Mark Watkins came up with the “compiler” ASSUMPTIOM that optimized the code, meaning the real Rybka code looks like Fruit, the ASSUMPTION is thus:

```
FRUIT                         RYBKA
if (doubled) {                if (doubled) {
opening[me] -= 10;               opening -= 0;
endgame[me] -= 20; }               endgame -= 158; }
```

Change the facts and the world according to the Rybka accusers is okay again.

If courts starts reasoning as the Rybka accusers then we
need to triple the number of prisons.

Last observation

We noticed this constant tendency of the Rybka accusers to use Rybka’s main defence, a different data structure (bit-board), against Rybka as **an excuse** to explain a difference in evaluation to incriminate Rybka further by calling it similar after all (!)

The norm should be simple, if Fruit evaluates different things than Rybka then Rybka != Fruit.

As an example see the following discussion, the evaluation of a backward pawn. Nothing special, standard chess knowledge, every reasonable engine has it. Then looking at what the code does is that Fruit uses the **Kmoch** definition of a backward pawn while Rybka doesn’t. Thus Rybka != Fruit.

Objectivity obviously has left the building.

On copyright infringment or GPL breach

Watkins on various places in his document hints to copy theft, we like to address these points.

1. setjmp (idea borrowed from Fruit by the admission of Rajlich to unwind the recursive search) (elo gain 0)
   Also found in other programs, for instance in Rookie by Marcel van Kervinck, TCP by Tom Kerrigan and dozen and dozen of other programs, see Dann Corbit's list, here. Meaning, not so unusual at all.

2. More serious is Watkins accusation in Chapter 6.3.1 Parsing the "position" string. Please find our analysis here.

3. We also researched Watkins accusation in Chapter 6.3.2 Time management. Please find our analysis here.

It was disappointing for us to learn that the error with "0.0" was **known** by Watkins one year before the ICGA investigation started (2010) but was withheld from his document, withheld from the Panel and withheld from every forum discussion long after the ICGA verdict. For many programmers (Wegner included) "0.0" was an hot issue confirmed by the hundreds postings on this issue alone.

More on 0.0

Robert Hyatt - I repeat, using a floating point constant when an integer arithmetic operation is being done is a programming error, except for specific and unusual conditions. Comparing an integer to a floating point constant is ALWAYS an error. ALWAYS. Never any POSSIBLE reason for doing that.

That is an error that was directly caused by COPYING fruit code. Where fruit DID use floating point variables, and a comparison to a floating point constant was correct. Or a floating point constant was assigned to a floating point variable was correct. NOT in Rybka, however.

But when looking at Crafty itself, it is full of such errors, see here. Meaning, programmers make such mistakes and aren't noticed because they are not harmful to the desired result.

Last: Also compare "Main()" between Rybka 1.0 beta and Fruit 2.1 or check Fruit "Main()" with Strelka "Main()" for a better visual. Usually in the business of detecting clones "Main()" is a good start to look at because its fundamentals and most of the time you will find strong similarities, even almost exact code as the Rybka 1.0 – Strelka case so obvious shows.

The evidence for copyright infringement or GPL breach already was extremely poor as one can not use a couple of code snippets to conclude *Rybka started its life as Fruit.* [Wegner] [Hyatt] and with the above listed counter evidence it would be praiseworthy when both gentleman to at least take those statements back.

EVAL-COMP comments

Note that Fruit uses a pawn value of 100 while Rybka uses a pawn value of 3200/3399 in EVAL which at the end of EVAL is transferred back to 100 for the usage in search. For the background of this idea see a forum post which explains the large Rybka numbers in case you might wonder. What was seen first as an obfuscation to hide the Fruit origin in reality is a nice (new) idea to tune the EVAL with much better rounded scores.

Note-2, when we use the term “abstraction level” it means we remove Rybka’s main defence (a bit board data structure) from the code and then look what is left, what the code actually evaluates.

Double Pawn code at abstraction level not similar.

```
FRUIT                      RYBKA
if (doubled) {             if (doubled) endgame -= 158
opening[me] -= 10;
endgame[me] -= 20; }
```

No opening | midgame code for double pawns in the Rybka binary. Thus Rybka != Fruit (even) at abstraction level.

Queen 7th rank code at abstraction level not similar.

```
FRUIT                      RYBKA
if (7th) {                 if (7th) endgame += 1420;
opening[me] += 10;
endgame[me] += 20; }
```

No opening | midgame code for Queen 7th rank in the Rybka binary. Thus Rybka != Fruit (even) at abstraction level.

Bishop Mobility not similar.

Various chess programmers have spoken out in several forums, we limit ourselves to the one that impressed us most.

Sven Schüle at Talkchess Engine Origins

I am on no “bandwagon”, Bob. Please avoid such wording, it is inappropriate since you are trying to downplay my contribution as if I were a newcomer involved for the first time. You know my position in the RF issue pretty well, for years.

I think that “Cray Blitz” is not very relevant in our discussion since it has the same author as Crafty, so I suggest that you avoid referring to CB. The points you raised about CB vs. Crafty were never put under question AFAIK but don’t contribute anything here.

Your notion of “calculating attacks on the fly” is different from mine. Either you look up information in a table, or you calculate that information “on the fly”, you don’t do both. The rotated bitboard approach as well as “magic bitboards” both include a “lookup” concept for sliding attacks, albeit in a different way. No need to explain that further to you, Bob. This is in contrast to other concepts like “dumb7fill”, for instance, where you really have “on the fly” calculation of attacks.

Having clarified the wording, I think we have the following situation (here is also the promised part of my reply to the parallel sub-thread where you replied directly to Ed):

Fruit 2.1:
– Feature definition includes not to count attacks to friendly pieces.
– Implementation is “mailboxy” using loops to calculate attacks “on the fly” and count mobility. This is quite expensive but there is probably nothing substantially faster for a mailbox program.
– Scoring is linear.
– Everything follows in a straightforward manner from “feature definition + board representation + scoring strategy”.

Crafty up to 20.1:
– Feature definition includes to also count attacks to friendly pieces.
– Implementation uses table lookup in a way typical for rotated bitboards. The precalculated information being looked up is the mobility count itself, which is only possible due to not excluding friendly pieces.
– Scoring is linear.
– The implementation is quite efficient within the conditions given by feature definition and board representation. It is also “straightforward”, since not doing a table lookup with rotated bitboards would make absolutely no sense.

Rybka 1.0beta:
– Feature definition includes not to count attacks to friendly pieces.
– Implementation uses table lookup in a way typical for rotated bitboards. The precalculated information being looked up is only the attack set.
– Scoring is linear.
– It seems to be most efficient under the given conditions which do not allow to store the mobility count in the lookup table. It is also “straightforward” for the same reason as in case of Crafty above.

Crafty 20.2 and later, pre-magic:
– Same as older Crafty above, with the exception of introducing non-linear scoring.

Crafty after switching to magic bitboards:
– Feature definition still includes to also count attacks to friendly pieces.
– Implementation uses database lookup typical for magic bitboards. The precalculated information being looked up is still the mobility count itself, which is only possible due to not excluding friendly pieces.
– Scoring is non-linear.
– The implementation is quite efficient within the conditions given by feature definition and board representation. It is also “straightforward”, since not doing a database lookup with magic bitboards would make absolutely no sense.

So the conclusion is obvious: in each of the cases above we see an implementation that is “typical” and follows in a “straightforward” manner from design decisions about
– the exact definition of the “mobility” feature to be applied,
– the given board representation (and the method of calculating sliding piece attacks which is related to it), – and the scoring strategy for the feature. All implementations are very different from each other, where of course the pre-magic Crafty versions are the closest ones. All implementations, perhaps with the exception of Fruit, are also heavily driven by efficiency issues, and appear to be most efficient under the specific conditions.

Therefore I see no reason to assume any “derived code” relation between the engines mentioned above regarding mobility evaluation, except “Crafty vs. Crafty” of course which we don’t need to discuss. All that is possibly shared between Crafty, Fruit, and Rybka is clearly on the abstract design level. But in none of the relevant cases there is even a sharing of the complete design, as shown above.

Conclusion, not similar.

2.6.2 Passed pawns, initial bonuses

Fruit evaluates only one square in front a passer to estimate how dangerous a passer really is, Rebel 3 squares, Rybka 1.0 does it all the way till the promotion square.

Another objection is that Rybka evaluates via tables, not via a formula.

Conclusion, not similar.

Objectivity

While we are talking about evaluation via tables note that Fruit has **only one** evaluation table, Rybka has many.

FRUIT: static const int KingAttackWeight[16] = { 0, 0, 128, 192, 224, 240, 248, 252, 254, 255, 256, 256 ,256, 256, 256, 256, }

RYBKA - from Zach Wegner's document:

```
const int shelter_value[5] = { 1121, 0, 214, 749, 915 }
const int storm_value[5] = { 0, 0, 2334, 653, 310 };
const int storm_value[5] = { 0, 0, 2334, 653, 310 };
int PassedOpening[8] = { 0, 0, 0, 489, 1450,2900, 4821, 4821 };
int PassedEndgame[8] = { 146, 146, 146, 336,709, 1273, 2020, 2020 };
int PassedUnblockedOwn[8] = { 0, 0, 0, 26, 78,157, 262, 262 };
int PassedUnblockedOpp[8] = { 0, 0, 0, 133,394, 788, 1311, 1311 };
int PassedFree[8] = { 0, 0, 0, 101, 300, 601,1000, 1000 };
int PassedAttDistance[8] = { 0, 0, 0, 66, 195,391, 650, 650 };
int PassedDefDistance[8] = { 0, 0, 0, 131, 389,779, 1295, 1295 }
int CandidateOpening[8] = { 0, 0, 0, 382, 1131,2263, 3763, 3763 };
int CandidateEndgame[8] = { 18, 18, 18, 181, 501, 985, 1626, 1626 };
```

And once when we addressed this obvious difference in programming style we got the usual **ASSUMPTION REASONING** from the Rybka accusers Rajlich was obfuscating the Fruit origin and this was nothing more than speeding up the Fruit code.

FIDE Ethics Committee slams ICGA board as ignorant of the law

FIDE complaint

On August 31, 2012 Vasik Rajlich author of Rybka filed a complaint to the Ethics Commission.

The sole focus of the complaint concerns the ethical standards of the ICGA process that led to the lifetime ban of IM Vasik Rajlich from ICGA events and the willful besmirching of his reputation.

IM Vasik Rajlich is a FIDE member and has been damaged and unfairly treated by an organization that is affiliated with FIDE.

This complaint contends that the ICGA or its appointees have breached several articles of the FIDE Code of Ethics, specifically:

2.2.2 – Office bearers who through their behavior no longer inspire the necessary confidence or have in other ways become unworthy of trust.

2.2.3 – Organizers, tournament directors, arbiters or other officials who fail to perform their functions in an impartial and responsible manner.

2.2.4 – Failure to comply with normally accepted standards of courtesy and chess etiquette. Misbehavior of a personal nature which is generally unacceptable by normal social standards.

2.2.9 – Players or members of their delegations must not make unjustified accusations toward other players, officials or sponsors. All protests must be referred directly to the arbiter or the Technical Director of the tournament.

2.2.10 – In addition, disciplinary action in accordance with this Code of Ethics will be taken in cases of occurrences which cause the game of chess, FIDE or its federations to appear in an unjustifiable unfavorable light and in this way damage its reputation.

Anyone acting in contravention of this code can be excluded from participation in all FIDE tournaments or from specific types of tournaments for a period of up to 3 years. Weight shall be given to the type of violation and to any previous violations in decided upon the length of the exclusion period.

View / Download the complaint to FIDE.

The FIDE ruling

The verdict came in a 17 page long PDF with (according to Dr. Levy) the following (external) confidentiality note from FIDE.

Certainly the outcome of the matter (as distinct from the detailed reasoning) is then a matter of public record even if the outcome would only be formally reported to the Executive Board or General Assembly once a year and the FIDE website would be updated from time to time. The Ethics Commission has no objection if the ruling is announced or even discussed publicly once released to the parties, but asks that a measure of confidentiality still be maintained on the basis that a copy of the judgment be distributed only to parties with an interest in the matter and not generally, and that the judgment not be published in its entirety on another bodyâ€™s website for consumption of all and sundry.

While we have received no such limitation from FIDE and see opposite evidence of published FIDE rulings all over the Internet (and on main chess sites) we (for the moment) will comply to above and limit ourselves to a couple of quotes from the 17 page document.

FIDE Ethics Committee slams ICGA board as ignorant of the law and breaching Vas Rajlich’s rights to a fair hearing. ICGA convicted of bringing FIDE into disrepute and requested to make amends and alter its statutes and procedures.

Chairman Mr. Roberto Rivello of the FIDE Ethic Commission
Jurist and Judge in Italy with jurisdiction on criminal cases
Professor of International Law and International Organization at the University of Turin

Quote-1
In occasion of the oral hearing in front of the EC, Mr Levy confirmed that ICGA has no rules on these issues.

However, in the documents submitted by the parties, ***quite strong and specific words were used, such as “tribunal”, “innocence”, “guilty” and “verdict”.***

In an article written by Mr Levy (together with Mr H.J. van den Herik, Mr A.Plaat and Mr D. Dimov) and submitted to the EC by the ICGA, the proceedings against Mr Rajlich are defined as follows: “The investigation, the report of the investigation, and the verdict that Rajlich was guilty of the plagiarism took place in the form of a version of Crowdsourced Online Dispute Resolution (CODR)”.

In another ICGA’s publication, the words “peer review” were used.
It seems clear that all these words and definitions do not correspond to their technical meaning. They have been used by analogy by persons who are not experts in law.

Quote-2
Members of the ICGA's Executive Committee ignored they had to respect specific rules if they intended to carry out disciplinary proceedings against Mr Rajlich. ***Ignorance of the law is no excuse, especially for persons charged of main responsibilities in a given organisation: they had and will have the duty to apply FIDE and national mandatory rules. However, they were not jurists*** and they were convinced to act in the best interest of their organisation.

Quote-3
In addition, Mr Rajlich was not informed about the existence of real disciplinary proceedings against him nor about the risk to be sanctioned this way. He was informed, and in a very informal way, only about the proceedings concerning "Tournament rule 2" and it is clear –from the exchange of messages between him and Mr Levy- that he was not fully acquainted with the possible multifaceted nature of the proceedings and his right to be heard. One person who is informed that his behaviour could be qualified as a cheating and sanctioned with a lifetime ban is likely to assume different decisions about his defence, in comparison to a person who just know about the risk to be disqualified from a tournament.

Quote-4
The EC unanimously rules that:

– Otherwise, by imposing a lifetime ban as a sanction against Mr Rajlich, in absence of a clear statutory basis and without sufficient procedural guarantees for Mr Rajlich, ***the ICGA did not act in accordance with FIDE rules, this way violating par. 2.2 and 2.2.10 of the FIDE Code of Ethics.***

– ICGA has to be sanctioned with a warning and has to be invited to modify their statutes in accordance with FIDE principles and rules.

The denial of a fair appeal

The double standard of the ICGA

In the past (the WCCC in Graz 2003) the ICGA banned a suspect program (the LIST program of Fritz Reul) because Reul refused to cooperate in an investigation. Later Reul did cooperate and Reul was reinstated after he handed over his source code to (expert) Dr. Chrilly Donninger who inspected the LIST source code and found it original.

So if Reul was granted an appeal one would expect Rajlich to deserve the same treatment especially now that a sea of contra evidence is available. But Rajlich was denied an appeal.

A poll among Chess Programmers showed that 16 from the 17 granted Rajlich the right of an appeal.

An USER poll showed that from the 30 votes 25 granted Rajlich the right of an appeal, 2 people voted undecided and 3 denied him that right.

Conclusions

1. There is hardly evidence that Rybka (according to Wegner & Hyatt) started its life from Fruit, the evidence for that is extremely poor, you cannot use a couple of code snippets to conclude that.

2. From the start (December 2005) Rajlich never denied he used many ideas from Fruit and it never was a problem as everybody else was doing the same.

3. As we have tried in this document to calculate the number of exact ideas in the version that actually played (Rybka 2.3.2) in Amsterdam 2007 and became world champion, the number is (only) 3 and maybe 5 maximum.

4. Taking 3-5 ideas can hardly seen as:

    - Taking too much.

    - A breach of ICGA rule #2