A CRITICAL ANALYSIS OF THE FOUR PARTS OF RIIS

MARK WATKINS, UNIVERSITY OF SYDNEY

I was asked by the ICGA to briefly address the 4-part article on the Rybka scandal that was authored by Dr. Søren Riis and recently published by ChessBase. As he elides (at least) one major aspect of the ICGA decision with Rybka, I will append a brief description of it at the end.

Additionally, about 3 months ago, I authored (in part to aid the Free Software Foundation) a 30-page PDF recapitulating the evidence behind the ICGA decision, and the process therein. However, having no reason to stir what were then (relatively) calmed waters in the world of computer chess, I did not make it public until Part I of Riis appeared. Here is a link to it.

0.1. **Setting.** ChessBase published a 4-part series from Dr. Søren Riis claiming that he had "investigated" the Rybka scandal. I don't know Riis other than from his role as a moderator on the Rybka Forum. His specialty is *theoretical* computer science, and it is not clear to me that he is even a programmer. This seems to color his ideas throughout. I will leave his credentials at that, and move onto his arguments. Here are links to his series: Part I, Part II, Part III, Part IV.

0.2. **About the author.** Mark Watkins is a mathematician at the University of Sydney, working on software development of the MAGMA computer algebra system. He is also working on an assessment of the current prospects for solving the game of Losing Chess.

1. PART I

Part I starts with a Summary, then gives an Introduction, a History of Rybka (with 3 games included), and then some description and comments about the ICGA Investigation. I will skip the Summary, as it is included in the rest.

1.1. **Introduction.** The Introduction states the purpose of the paper as a defense of the reputation of Rajlich, with Riis arguing that the ICGA's findings were misleading, and that the ruling lacked any sense of proportion.

The principal element of dispute here is this:

> Based on the evidence I will present, a person can form a very credible alternative conclusion: that the implementation of similar evaluation concepts and algorithms in a computer chess program will inherently lead to code similarities even if no code is copied. Thus, the statement *Rybka implements concepts and algorithms learned from Fruit* appears to be the most correct and accurate formulation...

Riis seems unaware that the ICGA process essentially considered his alternative formulation, and the consensus was that it was unsupported by the evidence. Rather,

the ICGA concluded that Rybka did more than just "implement concepts and algorithms learned from Fruit" – quoting page 5 of my above linked PDF, as a summary of part of the Rybka/Fruit evidence:

> The Panel concluded from `EVAL_COMP`, in part when viewed alongside other evidence, that it was quite clear that more than just "ideas" from Fruit 2.1 re-appeared in Rybka 1.0 Beta, but rather quite specific creative choices. Any individual element could be declared to be simply "Fruit influence", but the picture as a whole stretched much beyond that. There was a general consensus that the Fruit/Rybka situation was much beyond the "standard" amount of engine sharing that was typically permitted in author-based computer chess.

Riis continues:

> ...a semantically subtle difference that nonetheless completely overturns the ICGA's conclusion. It is important to understand that the implementation of ideas and algorithms learned from other programs is universal practice in chess programming as well as many other types of programming. The issue in contention in this case is whether source code was copied from one program to the next.

The standard adopted by Riis here "whether source code was copied from one program to the next" could only be said to be proper one when one says what this means. For instance, copyright law (cf. *Nimmer on Copyright*) can distinguish verbatim copying, transliteration, copying of a book plot (such as Hollywood adaptation rights), and more. The typical criterion used in copyright law is whether two works are "substantially similar", which is admitted (by Nimmer) to be the crux of any specific debate. Riis typically intends "code copying" to strictly mean an action in the style of copy/paste, a rather minimal view I would say.

From a different angle, copying is usually concluded to have occurred by inference from the end result, rather than via direct observation of the physical act of copying. The inferred nature of the offense is one reason why (say) copyright infringement is rarely a criminal matter, and carries only the burden of proof typical in civil matters. Riis contrarily appears to contend that if there is *any* possible counter-explanation for the observed Fruit/Rybka substantial similarity, no matter how far-fetched, and no matter that Rajlich chose not to offer such a defense, then the ICGA should not have acted.

I bring up copyright law here as it formed a framework for many of the issues considered by the ICGA Panel. I will expand a bit on this in Part III, though here I might note that, following van den Herik and Swiercz in their ICGA Journal editorial on the matter, any aspects of copyright law were used merely a guide, and the ICGA need not adopt all of its peculiarities.

1.2. **Rybka history.** After insinuating that the ICGA "attempted to ruin [Rajlich] on a world stage", Riis then shifts to a History, presenting three games of Rybka, and claiming (as has been said about many a top engine over the years) that its insights were sensational, etc. There is not much to say here, but one minor correction: he states *Rybka went on to become a commercial engine in 2006*, when it was already commercial in December 2005 (one can cite CCC posts from that month which speak of payment being made, but the download yet to work).

1.3. **The ICGA investigation.** Riis first gives his (outsider's) history of the ICGA investigation of Rybka. He then quotes ICGA's Rule #2 for the WCCC, noting that it has been invoked on a number of occasions, most notably in 2006 when LION++ was disqualified. This is notable not only because LION++ was disqualified for being a Fruit-derivative (they even acknowledged this, but did not state it on their entry form), but also, as noted by Levy in his piece *A Very Sad Case* in ICGA Journal 34/2, because Rajlich was present at the event in Turin, and thus was necessarily aware of the ICGA interpretation in all aspects therein.

Riis then quotes Rajlich (I do not know the source) as saying that: *When two modern top chess programs play against each other maybe 95% of the programs are algorithmically the same.* Indeed, Riis hectors his audience into agreement here, saying that "to begin to understand" his argument we "can start by acknowledging the truth contained" therein. Unfortunately, Rajlich's statement, unless couched very carefully (such as taking "algorithmically" to exclude choices in the evaluation function), is simply not true. I doubt that one can find many programmers who would agree with it in the manner employed by Riis. Indeed, Rajlich himself stated in a post from December 2005: *As far as I know, Rybka has a very original search and evaluation framework.* How this is to be reconciled with the 95% numerology remains obscure. This is the second major point of dispute in Part I, and one that Riis does not begin to start to investigate further.

Riis then concludes, mostly from his acceptance of Rajlich's statement, that Rule #2 is obsolete and absurd. He then turns to what I would term a "sophomoric lawyering" mode, where he queries *exactly* what various terms mean. A better method to resolve such pedantry would be an inquiry to the ICGA upon entry, or perhaps at the mandatory meeting before the event. If the issue is sufficiently broad, a Communication in the ICGA Journal (asking for feedback from the members) might be superior. Indeed, the failure of the entrant to take affirmative measures toward a common understanding was precisely what caused such a kerfuffle in the LION++ case previously noted by Riis.

Answering his questions:

(i) "Original" has been interpreted by courts to mean where something has its origins. The ICGA process determined that the "origins" of Rybka, particularly the evaluation function, were in Fruit.

(ii) Said programmer should *ask* the ICGA for guidance.

(iii-iv) It is prerogative of the ICGA to interpret its rules. See also (ii).

(v-vi) (v) No, there is (currently) no necessity. (vi) The clock for a statute of limitations typically only starts when an offense is discovered.

(vii) I do not know about safeguards. However, based on my correspondence with 7 of the entrants in 2007 (for example), all of them understood the rules in a manner that led them to conclude that Rajlich violated them.

2. PART II

Part II has five sections, the first about a purported paradigm shift in computer chess, the second concerning the fact that Rybka has been reverse-engineered, then one about "plagiarism", one about similarity testing data, and finally Riis queries what "original" might mean.

2.1. **Paradigm shifts.** Riis much ballyhoos that the open-source release of Fruit in 2004-5 started a paradigm shift in computer chess, then argues that this has implications for Rybka originality. He gives Elo-based data to support his contention. Unfortunately, the data seem erroneous at many points, and even when correct they appear to be interpreted in a suspect manner.

To allow the reader to skip this section, let me say that Riis could have alternatively argued (without contention) that Rybka's gain of 85 Elo/year from Dec 2005 to Aug 2008 (Rybka 3) was unprecedented for a #1 engine. Then, rather than haunt his data, I would merely have noted a few possible external influences on this (e.g., that multicore testing regimes became popularly available about that time), while readily giving due to Rajlich's (and Kaufman's) skills.

Returning to his argument, Riis gives a graph that he claims derives from historical SSDF and CCRL data. It consists of three engines from 1992-2003, and then eight more from portions of 2004-2011. He does not say how he adjusts for hardware differences in the SSDF data, which CCRL time control he uses, etc. Furthermore, some of the data points seem to come from other sources. For instance, the data points given for Rybka, Fruit, and Naum in 2004 cannot come from either SSDF or CCRL, as no relevant version is included in their lists. Next, the datapoint for Rybka in 2005 looks to be near 2675; but the SSDF number for Rybka 1.0 Beta (Dec 2005) is 2773 (on slow hardware), and CCRL puts the number closer to 2900. Similarly, the Fruit datapoint for 2005 appears on his graph at a bit above 2700, while CCRL lists Fruit 2.1 (Jun 2005) about 75 points higher.

Though I am just guessing on the methodology used to construct the graph, it seems that Riis also rounds off years, so perhaps Jan 2005 and Dec 2005 would both fall under 2005. This could lead to rather clunky wobbles when (say) an engine releases a version every 18 months. Engines that release multiple times in a year (such as Stockfish) could also have their data skewed.

Riis then zooms in on the 2004-2011 era, and claims that during this period, engines tended to increase in Elo faster. This compares apples to oranges (or pears) in various regards. One issue (among others) here is that he has conveniently included "early development" datapoints from his dubious 2004 data; this is additionally an issue when splicing SSDF/CCRL data into a historical growth rate picture, as the former would typically wait for an engine to reach a rather high level before entering it into their list, while the CCRL tends to accept engines at earlier stages of development.

Riis then gives a table that highlights Rybka 1.0 Beta on a 64-bit machine compared to Fruit 2.1 on a 32-bit machine. He states: *from first release Rybka was already far ahead of Fruit, and the gap just kept widening.* I'm not sure what "Fruit" means here, but Fruit 2.1 was sort of a development snapshot, and indeed Letouzey himself added about 75 Elo to it by the end of 2005 (via both simple "engineering" speed-ups, and other more creative aspects). Incidentally, Fruit was commercialized in late 2005, and Letouzey was about to sign a contract with Convekta to license Fruit to be packaged with their GUI – but then Rybka appeared, and talks ceased.

Riis then notes that HIARCS "had its biggest Elo jump in twelve years" (with HIARCS 10) about 6 months after the Fruit 2.1 source code appeared. However, this datum from the HIARCS webpage is *version-based*, while Riis is arguing on a *time-based* metric. Indeed, there was a gap of more than 2 years between HIARCS 9 and 10, somewhat longer than typical, so it might not be surprising than the Elo

gain was (slightly) higher than normal. Furthermore, with a bit of further investigation, Riis might also have found that HIARCS 10 was the first version for which Mark Uniacke was working full-time.

Riis then turns to Junior, but his data here seem bogus. The aforementioned graph puts Junior in 2005 at about 2800, and around 3000 in 2006. I have no idea how he gets these numbers. The CCRL 40/40 data lists Junior 9 (late 2004) at 2778, and Junior 10 (Aug 2006) at 2843. This is about a 70 Elo gain over a period of more than a year and a half, fairly near his expected rate.

On the whole, even if the data were correct, I still don't see the relevance of this argument; Riis acts as if a sufficiently large Elo gain suffices as a sign of originality, but it would seem that it only implies **partial** originality. In this regard, the ICGA never disputed that some/many parts of Rybka were original, and indeed one suggested disciplinary action was simply to change the relevant victors to Fruit/Rybka and Rajlich/Letouzey; due to Rajlich's non-desire to interact with the ICGA, this was deemed unviable.

2.2. **Reverse Engineering of Rybka.** Riis then gives a narrative of how Rybka has been reverse-engineered. Some of his historical claims are disputable (such as how well accepted Strelka was), and, e.g., he seems to erroneously place the caper of "larcenous intent" in mid-2008, whereas Kaufman has posted that this occurred in early 2007. Riis asserts that many other authors have copied all the stuff thus learned from Rybka, though he does not specify how widespread this is; for instance, his Elo graphs for Junior and HIARCS show no apparent rating spikes corresponding to any information derived from Rybka disassembly. He then chides the ICGA for acting against Rajlich *in absentia*, failing to note that this was the latter's own choice. He concludes by claiming that the ICGA may have committed an illegal act in publishing a Rybka 2.3.2a disassembly, ignoring that the relevant parts are derivative of Fruit.

2.3. **Plagiarism.** Riis then turns to the question of "plagiarism", stating that Rajlich's various comments of praise toward Letouzey suffice to overcome this charge. After giving three examples of such praise, he proceeds:

> By definition, plagiarism *only* happens when credit to sources is *not* given, which was never the case with Rybka. . . If these acknowledgements were insufficient to satisfy WCCC Rule 2, then we must return again to arguments made earlier on the inadequacies of the rule itself and how it is applied.

However, as already seen in the LION++ case cited by Riis, some acknowledgement **on the ICGA entry form** is necessary too; and as noted by Levy, the presence of Rajlich in Turin in 2006 surely made him aware of this expectation. Thus any claim that Rybka never failed to credit its sources is mendacious, as Rajlich did exactly this by not *directly* citing them with his WCCC entries.

Furthermore, even making a proper acknowledgment of sources would not suffice for WCCC entry, as Letouzey himself would have to approve this use of his work (as he did with GridChess in 2007 and ClusterToga in 2008). Riis seems unclear on this point (in line with his claims of the inadequate nature of Rule #2), perhaps since he thinks there there was only some minor "Fruit influence" in Rybka, whereas the ICGA process concluded that there was much more than this, to the extent that Rybka could not be considered to be an original work of solely Rajlich. Similarly,

Riis thinks a few words of generic praise of Letouzey should suffice; however, this would appear rather disproportionate when balanced against the amount of Fruit-derived material contained in Rybka.

I'll finally say that I also think Riis makes too much of word "plagiarize" on the whole here; note that the ICGA decision says: *Vasik Rajlich is guilty of plagiarizing the programs Crafty and Fruit,* **and** *has violated the ICGA's tournament rules.* Riis appears to identify these two clauses, thinking that Rajlich's rule violation (of nonoriginality) is identical with whether or not "plagiarism" occurred (e.g., failure to properly cite sources). One cannot enter someone's else work into an ICGA event without appropriate permission, even if it is properly cited.

Riis then tries to position Robert Hyatt as the sole mouthpiece of the ICGA, even while admitting that (by its statutes) only its President David Levy can speak for it. I don't know whether this is an attempt to try to control the conditions of debate, or is just a corollary of my impression that Riis has done little reading about the issue outside the Rybka Forum (where Hyatt has made several thousand posts). For instance, I would be most happy if I could take the writings of Riis in lieu of a defense of Rajlich himself, but I doubt that either of them would find that to be proper.

2.4. **Similarity testing.** Riis then turns to "similarity testing" of computer chess programs. He does not much justify the relation of this to ICGA originality other than to quote Rajlich. He attempts to analyze and propound the extant data, largely (in the end) relying on the findings (and opinion) of others such as Ballicora. Again I find the issue to be tangential, and so will simply quote a post by Adam Hair on the issue:

> The phylogenetic tree that was developed from that data did not have Rybka 1.0 in the same branch as Fruit 2.1... **However**, this does not mean that the relationship between Rybka 1.0 and Fruit 2.1 is no closer than between two unrelated engines... As can be seen in the graph, the percentage of matched moves between Rybka 1.0 and Fruit 2.1 is not spectacularly high... but is significantly different than the average. While not necessarily condemning Rybka 1.0, this data **does not** exonerate it either.

I myself have never understood the conclusive relevance of such similarity tests. The developers of the similarity tool themselves admit that it is best used to determine which engines to investigate further, not to make conclusions that are more adequately addressed by more direct methods.

It is also to be noted that lack of (move) similarity does not necessarily imply originality. Indeed, even the converse can be true, for there is no rule against trying (abstractly) to match the moves of others. Famously, Deep Thought tuned its evaluation to match moves played in GM games. Furthermore, a similar dendrogram (albeit, with some additional caveats as of yet) to the one highlighted by Riis was produced by Laskos, and showed Fritz 11 to be "definitely" in the Rybka/Strelka family, so much so that it was declared to be a Rybka clone; however, code analysis of Fritz 11 appears to exonerate it, and one current speculation is that the Fritz team attempted to match Rybka moves via a fully legitimate method (such as mentor assistance, as described in this link).

Riis is hopeful that such similarity testing could yield a robust measure of originality. Though I find the results from these tests to be interesting, I contend there are too many problems for them to be used blindly, particularly when there is more direct evidence available. For instance, one can presumably pilfer 25% of the someone else's work, and yet pass under the radar. If such (partial) misappropriation were then discovered at a later date, I would think it quite odd for the indirect similarity data to override a conclusion from direct code analysis. In short, I find this section a diversion.

2.5. **Originality.** Riis concludes Part II by querying what defines an original program. He gives a quotation from Letouzey that is asserted to come from a 2008 interview. This is erroneous; the interview is from the first half of 2005, as can be seen from the fact that Fruit 2.0 is termed the most recent version. Indeed, this error is of pertinent import, as the evaluation function in Fruit 2.1 is somewhat more developed than that of Fruit 2.0 (for instance, adding KingAttack logic, open files for rooks, pawn storms, and more), and thus the quotation from Letouzey is (at the very least) deprived of proper context via its erroneous dating by Riis.

Riis concludes by opining that evaluation terms (for instance) should not actually count as "game-playing code" as Rule #2 is written. This ends up being more related to his next Part, but here I might note that by overstressing this single phrase, he tends to diminish the preceding sentence in Rule #2, which reads simply: *Each program must be the original work of the entering developers.*

Riis herein again seems to suggest that Rajlich only copied "ideas" rather than "code", while the ICGA concluded otherwise. My impression is that he intends the word "code" to be construed quite literally here, while as noted above, (e.g.) copyright law takes a significantly broader view. But this will be seen more in the next parts.

3. Part III

Part III turns to a small portion of the actual evidence of Fruit copying in the Rybka case; if nothing else, this is a welcome break from disputes over the meaning of words.

Riis first talks about evaluation in Rybka, listing 10 significant differences from Fruit; I found at most 4 of the 10 to be valid as such, and all were already noted by the ICGA process. He then dismisses the (crucial) "evaluation feature overlap" data largely via the ruse of pretending that one can re-vision the evaluation function as consisting of "concepts", each of which is so over-specified to have essentially one implementation, and then proceeds to advise that copying such "concepts" is not forbidden. He concludes with the issue of the floating-point zero.

3.1. **Evaluation.** The first principal claim of Riis is that Rybka and Fruit are grounded on different valuation conventions. He asserts that Fruit values a pawn as being worth 1.00, but in fact Fruit 2.1 values it as 0.80 in the opening, and 0.90 in the endgame. Riis then states that Rybka's evaluation was based upon "projected winning percentages", not giving any details.

I am left to wonder what this means. Having perused the Rybka internals, I see no real sign of this. Additionally, Riis himself admits that Strelka 2.0 is quite close to being a clone of Rybka 1.0 Beta, yet this "projected winning percentages" element appears nowhere in it either. At best, I can only suspect that Riis is indicating

an overall renormalization, in that Rajlich wants 1.00 to mean (say) 85% winning percentage throughout the game, rather than one pawn up (which might indicate a differing win% as the game progresses). I don't see any evidence that this is done in any constitutive way *inside* the Rybka evaluation function, but perhaps can be said to occur at the final interpolation between opening and endgame values. If this is such a "big evaluation difference" as claimed, I would expect it to have more visibility than just post-processing.

Riis then enumerates five major differences between Fruit and Rybka, given by Schröder. Of these, the first (lazy evaluation) is something that can easily be wrapped around existing code, and thus does not contradict the fact that other parts of the Rybka evaluation code are Fruit-derived. The second (futility pruning) is not an evaluation item; I would classify it as belonging to search. It is well acknowledged that the searches of Fruit 2.1 and Rybka 1.0 Beta differ, though both stay fairly close to classical depth-first iterative alpha-beta PVS (this is not so true of later Rybka versions). I am not sure what Riis means in his third point. He is not kind enough to indicate where this appears in Schröder's work, and the latter spans across many webpages. In any event, Rybka sometimes uses pre-computed arrays of numbers in correspondence to information that Fruit computes on-the-fly (as in his next point). The final two "major differences" are merely procedural in nature; they again do not affect any measure of originality in the creative aspects of the evaluation function.

Riis then states the above six differences are "actually substantial and impact playing strength" – I claim: projected winning percentages is not substantial, futility pruning is not evaluation, the KingSafety point is obscure and, as with the last two points, looks to be merely an engineering improvement (indeed, Letouzey himself made similar such speed-ups when preparing Fruit for the 2005 WCCC). I would say that only lazy evaluation could be considered "substantial" – in the ICGA discussion, this was considered as one of about 40 "evaluation features" that typically appear in chess engines.

Riis then quotes Rajlich concerning "four more big differences" (in evaluation) between Rybka 1.0 Beta and Fruit 2.1. The first, Rybka's material imbalance table, is quite famous. This is based on Kaufman's investigations, and Rajlich had this in Rybka already in 2003. The ICGA discussions again noted this as one of the evaluation feature differences. The second, passed pawns, is mostly correct. Fruit 2.1 has an overall yes/no criterion for whether the square directly in front of a passed pawn is occupied, and if not, whether the pawn can advance one square safely (when a bonus is given). Rybka 1.0 Beta splits this into three conditions: whether a friendly piece is in the pawn's path, whether an opposing piece is, and whether or not the pawn would be attacked but unguarded when advancing. A bonus is given for each of these separately. Furthermore, the entire path in front of the passer is considered, not just the immediate square in front. While I agree that Rybka's method differs from Fruit overall, there is still a partial sense of similarity. The EVAL_COMP analysis (§2.6.4) gave Fruit/Rybka a 0.3 overlap on a scale of 0 to 1.

The third, tuning, brings up the important what/how-much issue. Namely, an evaluation function has two parts: *what* to evaluate, and *how much* weight to give each feature. Again the ICGA process recognized that Rybka had indeed varied the latter; the complaint was that Rajlich copied Fruit in the former. Furthermore, there was a general feeling that the "creative aspects" of the evaluation function

were more embodied in the choice of *what* than *how much*, and indeed in 2005, the latter might even be something that one could out-source to various black-box programs (additionally, Letouzey himself applied some tuning to Fruit later in 2005). The fourth difference, where Rajlich claims Rybka does not score doubled pawns, is only partially correct concerning Rybka 1.0 Beta (Rajlich himself admits doubt). Rybka (like Fruit) interpolates opening and endgame scores throughout the game; the 1.0 Beta version has no doubled pawn penalty in the opening, but does have one for the endgame. Thus, as soon as any piece is removed from the board, the penalty becomes nonzero.

Of the "ten substantive evaluation differences" listed by Riis, only three remain (four if one includes tuning); these were all noted in the ICGA process. One could append a few more Fruit/Rybka evaluation differences (such as recognizers for specific endgames), but overall any accounting must be contrasted to 20 or so evaluation features that Rybka and Fruit essentially have in common, typically with rather particular details. Other engine comparisons showed much greater diversity in their feature choices.

Riis then shows us a screenshot of 17 Microsoft Word documents claimed to be written by Rajlich in 2003-2005. After effusing about the wonders therein, Riis concludes: *It is exceedingly hard to see the point of developing a slew of original ideas for Rybka only then to copy Fruit's evaluation.* Riis seems unaware of how many "original ideas" in computer chess in the end simply do not work; or, to answer his implied question: **because the Fruit-based evaluation tested better**!

Riis ends this section by saying that there is a common misperception that Rybka 1.0 Beta did little but transcribe Fruit's evaluation. To show a possible source of said misperception, he adduces quotations from Wegner that tended to exaggerate the Rybka/Fruit situation. For instance, Wegner says that *Rybka's evaluation is virtually identical to Fruit's*; this can only be true at the "what" level of evaluation features, as Rajlich did indeed tune the numbers. And as noted above, Rybka does add a few elements such as lazy evaluation and material imbalances. Other than this, I mostly agree with Wegner. Similarly with the second statement; other than in how backward pawns are delineated, and again with the caveat that Rajlich tuned the numbers, the pawn evaluations (albeit perhaps of low information content) are substantially similar. Finally, I personally find it completely clear that Rybka 1.0 Beta used the PST framework of Fruit 2.1. One can recover the exact Rybka PSTs from the Fruit 2.1 code with 17-18 changes of numbers (tuning, as before), and changing a mere 4-6 lines of code (all but one being an adjustment of the score for a specific piece-square). No other examined engine required such a small set of changes (firstly even assuming that it had the same PST scheme). Notably, even Fruit 1.0 differed significantly from Fruit 2.1 here. Given that Riis has not yet mentioned PST at this juncture (he does below), it is rather unclear why we are to think that Wegner is wrong here.

3.2. **Feature Overlap.** Riis proposes to describe each evaluation feature in sufficiently specific terms so that there can essentially only be one resulting implementation – yet he wants this description to still be just an "idea". This is, in essence, a rather standard dodge to get around copyright law. In rebuke, courts have noted that if an "idea" (or concept) is so minutely described to have only one realisation, then it typically is not an "idea" anymore. Riis chooses a bare example (rook on 7th rank), and displays this as proof that there can only be one definition. Of course,

not all evaluation concepts are so explicit; e.g., precisely what should one compute to determine whether a bishop is bad? But even the example given by Riis fails his purpose: some engines give a rook on 7th bonus all the time; some only do so when the opponent's king is on the 8th or an enemy pawn is on the 7th; some add a bonus for doubled 7th rooks (perhaps depending on whether they mutually guard each other), some for a rook and queen both on the 7th; some have a similar bonus for a rook on the 8th (or even 6th), some don't (see §2.3.5 of EVAL_COMP). One sees that even in this basic example there is room for creativity, which is greatly multiplied across the 20-40 features a typical engine possesses.

Furthermore, even if the above were legitimate, Riis seems unaware that a collection of uncopyrightable things can itself be subject to creative protection in its collative aspects. Therefore, a typical comparison for the purposes of substantial similarity will use levels of abstraction. For instance, with a novel one might have: *a love story, between youngsters, from rival families, etc.* As Nimmer indicates, eventually one gets a plot that infringes upon that of *Romeo and Juliet* (he suggests that *West Side Story* already does so).

The ICGA process, at the suggestion of Wylie Garvin, thus applied in its analysis a adaptation of the Abstraction-Filtration-Comparison test, and concluded that at the feature level, Rybka and Fruit were substantially similar. Though he might not like my phrasing, I would say that Riis dismisses the entire analysis by *de facto* arguing that there is no creative content in an evaluation function. But then, I might pose to him: what would stop someone from re-implementing the Rybka evaluation function, claiming that every part was merely an idea or concept?

Riis then notes that many EVAL_COMP engines were weak, but fails to note that a later analysis by Adam Hair, while confirming a rather weak correlation between features/strength, demonstrated Fruit/Rybka still to be a great outlier. Additionally, as noted in my recent recapitulation PDF, eliminating the weakest engine (Faile) makes the Rybka/Fruit overlap even more of an outlier. Riis here also seems to suggest that all world-class engines use essentially the same evaluation features. The ICGA Panel considered whether this was likely to be the case, and found it (often from personal experience) to be woefully implausible. As but one example, Rybka 2.3.2a has a Fruit-based evaluation function, while Rybka 3 has one re-written by Larry Kaufman. Both were world-class engines, yet their evaluation functions are markedly different.

Riis continues by noting the analysis was "subjective". Unless one has a phobia about the matter, subjectivity arises almost everywhere in life, such as in this debate, and certainly in a quasi-legal process like that of the ICGA. Other than voicing this as a philosophical scare word, I can't see that Riis has any complaint. Furthermore, one could hardly say that the analysis is *arbitrary*, as it is accompanied by 50 pages of descriptions; a reader can judge the current "subjective" debate between Riis and myself by the force of our arguments, and can do the same with EVAL_COMP.

In his last paragraph Riis pretends there is an issue of "data interpretation" with the raw percentage overlaps. The method to interpret these is given in the ICGA documents. Namely one applies a (standard) statistical analysis to the raw percentages, and measures the variance of the Rybka-Fruit pairing against the control group. This should be fairly routine for anyone with a mathematical background, and moreover is indicated in both EVAL_COMP and the ICGA Report. Thus I find

the query of Riis here to be rather odd, almost sciolistic in nature. At any rate, one concludes that the Rybka/Fruit data point was an outlier at 6-8 standard deviations in a pool of around 30 comparisons, or more than a 1 in a million chance of occurring at random.

3.3. **The floating point zero.** Riis erroneously claims the "floating point zero" is a *major* issue for ICGA; this is simply fallacious. Riis makes no attempt to justify his audacious claim (complete with italics), but I will rebut it anyway.

For instance, the floating-point zero issue forms part of one paragraph of my original `RYBKA_FRUIT` analysis (with the representative code sections), this itself being but one of the issues brought up in the Time Management subsubsection of a section specifically labelled *Things of lesser importance.* Similarly with the other main general sources of Rybka/Fruit evidence: Wegner does not mention it in his work; I don't think it came up anywhere on the ICGA wiki; and in my recent 30-page recapitulation, I mention it almost in passing, as a 6-line introduction to a larger code sample that contains 2 or 3 other common Rybka/Fruit points (in time management). Thus I don't know how one could **remotely** conclude it was "major" issue, as Riis assures his readers. Riis has the chutzpah to say that it is "no joke" that this was a *major* issue for ICGA; I propose that this comment is a derelict sign of his gross and utter inability to comprehend the debate.

Rather than argue with Riis over whether a typo (or maybe half of one) is or isn't relevant, below I will briefly enumerate some of the voluminous Rybka/Fruit material that he leaves untouched. Indeed, he deftly ignores the bulk of the ICGA evidence presented, and hankers upon the 2 or 3 items that best fit his purpose. As these few tidbits seem to form the major talking points of the Rybka Forum, I am left to wonder if Riis is even aware of the the rest of the evidence. I refer the interested reader to pages 14-29 of my recapitulation PDF for a brief synopsis of the Rybka/Fruit copying evidence.

Riis then states that time management is not game-playing code. He bases this *outré* notion on the fact that both Fruit and Rybka mix their time management code with the code for UCI parsing. However, most other engines separate the two – should Riis thus think that their time management *is* game-playing? Indeed, this idiosyncratic mixing of UCI parsing with time management was one of the (many) signs that suggested Rybka did indeed have its origins in Fruit; furthermore, one can note (see my §5 below) that the 2004 Rybka had a completely different UCI implementation, with the 2005 version curiously adopting Fruit's structure.

Riis concludes by comparing UCI parameters, noting that Rybka lacks about 18 of 20 parameters possessed by Fruit (and adds a SearchDirection parameter). Again this point seems bizarre; a minimally competent programmer should take no more than 15 minutes to excise such parameters from the Fruit code (making them their default values, if nothing else). And if such an accounting gives a sense of whether two engines are "markedly different", by what measure does he (as in Part II) conclude that, say, IPPOLIT is derivative of Rybka 3?

3.4. **Rybka/Fruit copying that Riis omits.** Here is a brief enumeration of a few of the points not covered by Riis. As noted above, these are expanded somewhat in Section 4 of the recapitulation PDF.

    (i) The Rybka iterative deepening code is largely the same as Fruit's, adding only a few elements (like mate scores), and varying a few parameters. The

semantic similarity is high (both check a variety of conditions in the same order), and the six variables of interest are allocated identically. I don't know of another engine that uses the general Fruit concept here, let alone one with so many code congruences.

(ii) Rybka's root search follows the same pattern of that of Fruit, including more than 5 notable points of commonality. An examination of other engines showed none that followed said pattern.

(iii) Rybka copies a small piece of search control code from Fruit, where again the ordering and variable allocation are the same. The Fruit code also includes some idiosyncratic redundancies, which curiously re-appear in Rybka.

(iv) The hashing structure of Rybka is the same as Fruit's, except for transposing bytes 8-11 with 12-15. Again I cannot find another engine with anything so similar; indeed, already Fruit 1.0 differs.

Some of these are more important than others, and a few more can be appended; I think the overall picture is clear. Furthermore, the 2004 Rybka version (see §5 below) had none of these Fruit commonalities.

4. Part IV

Riis starts his final part by describing the PST issue, intertwining this with an evisceration of Robert Hyatt. He then talks about some coding techniques of Rajlich, asserts that Rajlich was working continuously on Rybka for longer than the ICGA assumed, defames the ICGA process, and concludes with his subjective view.

4.1. **PST.** To pre-empt Riis (as he diverges rapidly), the evidence here is clear. As I stated above, the Rybka 1.0 Beta PST can be generated from the Fruit 2.1 code by changing 17-18 numbers (tuning), and 4-6 lines of code (amongst about 50). In the hopes of defogging the morass through which Riis wades, I challenged anyone to find any other engine (not Fruit-related) whose PST could be derived from such a small set of changes from the Fruit 2.1 code. There has been no answer.

Riis starts by quoting Rajlich on the practical importance of PSTs, claiming that "any reasonable choice" leads to no more than a 1 Elo difference. No evidence for this is given; the most relevant datapoint of which I know derives from Hyatt, who stated that 6 Elo is lost by completely removing the bishop PST in Crafty. Again I find Riis to be too accepting of Rajlich's claim, at the least.

Riis then claims that PSTs should use small numbers, to be efficient. Given that the Fruit PSTs are computed only once at startup (and in less than a millisecond), while the Rybka PSTs appear to be pre-computed into a table in a compilation phase, I simply have no idea why efficiency should be of the slightest import.

Riis then states that the final PST tables of Rybka and Fruit differ; however, the noncongruence of raw numbers was never the point. One reason for this is that Rybka has a different internal scaling, and another is the presumed tuning that Rajlich did. As above, after excluding the latter, there is a minimal amount of code change to go from Fruit to Rybka in their PSTs, much less than other engines. This is the metric of relevance. Riis then claims that Ballicora "persuasively" argues for their difference, but his first link merely has a marginally related brief sentence about semantic equivalence, while the second just lists some raw output,

mis-interpreting what the word "match" (of Hyatt) should mean in the context. Though termed "demolished" by Riis, the PST evidence stands firm.

4.1.1. *More PST.* Riis then claims that the presentation of PST evidence as given by the ICGA report was misleading, with "tables of virtually identical side-by-side Fruit and Rybka code" subject to mis-interpretation. I don't think he means the ICGA Report (of the Secretariat) here, but Wegner's document wherein he reconstructs "Rybka code" that matches the Rybka executable, and does so in such a way to show how little it need differ from Fruit 2.1 (in my RYBKA_FRUIT document I described this PST evidence using template formulas). Wegner is quite clear in his document that: *The code shown here is simply the functional equivalent; it calculates the Rybka PSTs.* I would think that anyone conversant with code disassembly should understand this point immediately. Panel members in particular would know that they were not dealing with any actual source code of Rybka.

Riis then makes more erroneous claims about the Panel process. For instance, he over-estimates the PST evidence as being the "most damning portion" of the report, when in fact it does little more than appear as one element (among 7) in a "partial listing of Fruit identical terms" (from evaluation) derived from Wegner's work, whose enumeration is but one page of the 14-page Report. Again Riis seems dreadfully misdirected in his sense of proportion. Finally, his inability to distinguish Wegner's document from the ICGA Report continues: *To support this the report provides page after page of near-identical source code side by side.* But this appears only in Wegner's work (which covers all the evaluation, not just PST).

Riis then points out that no actual Rybka source code was used by the ICGA. While the Panel discussed this issue, and might have found any such source code to be useful, disassembly proved a suitable surrogate in the end. Riis claims that anything so obtained was not "original Rybka source code", and thus should be removed from consideration. However, substantial similarity between Fruit and Rybka was not shown at the source code level, but at a level of abstraction suitable for comparison. Again, anyone conversant with the issue of re-constructing code in the absence of source will see this. Riis suggests that the ICGA (and Panel members such as Ken Thompson) misunderstood this issue, while I contest that the confusion is his. For instance, under his proposal, it seems that since Rajlich has discarded the source code, no one can *prove* he copied it, and thus he must be found not guilty. However, in the real world, the law cannot be mocked as such.

Riis then gives a picture of Wegner's PDF, largely retracing the ground I covered in paragraphs previous, with his typical bleating: *Since no one has a copy of the Rybka 1.0 Beta source code, no one can prove otherwise.* Why Riis spends so much time on such dogmatism is beyond me. He then impugns Robert Hyatt for a few paragraphs, yet again conflating Wegner's PDF with the ICGA Report (having previously asserting that Panel members almost certainly based their opinions merely on the latter). For someone so concerned about how misleading the Report (with Hyatt as one of its authors) might have been, Riis seems to be quite casual in his knowledge of the precise contents therein.

Unsatiated by his previous rumblings against Hyatt, Riis produces another section of this, including his quirky inability to construe the word "code" contextually. About the only thing worth noting here is his admiration of Ballicora's reconstruction of the Rybka PSTs using non-Fruit code. This seems unremarkable to me, and indeed, given nothing but the raw Rybka output, most CS students can do the

same. However, if one were to give said students both the Rybka output **and** the Fruit 2.1 code, I suspect that many of them would adapt the latter rather than work from scratch. This again shows a typical failing of Riis: rather than examine the evidence and apply some principle like Occam's Razor and/or inference concerning the most likely scenario, he huffs on about not being able to *prove* that something didn't happen in some obscure way.

4.2. **Language issues.** Riis claims that Fruit is written in C. This is technically false, as it is written in a very C-like C++. For instance, Fruit has typed constants (such as ColourNb), but neither classes nor templates. Riis then asserts that since Rajlich claims to have written his PST generation code in C#, "this means there is a 100% certainty that Rajlich did not copy the Fruit PST generation code." Here Riis seems unaware that C/C++ have significant overlap with C#, and code for them will often compile in C# with minor changes (a relevant one for Fruit's PST would be moving global variables to a class). Furthermore, he construes "code" in a quite literal sense; however, even translating the Fruit code into a less related language such as Pascal would still be an infringement, let alone making a few minor tweaks with C# nuances.

Riis then notes that Rajlich claims to have used `#define` in a different way than Fruit, again effusing over what appears to be a standard programming technique (or task), that of macro creation/usage. The overall point is obscure, and Riis himself says this *does not necessarily constitute a defense against code-copying.* He continues to be quite rigid in his copy/paste notion of the latter.

4.3. **The opportunity window.** Riis chides the ICGA report for failing "to mention that Rajlich became a full-time chess programmer in 2003." The most trenchant reason for such an omission is that the statement is false. Rajlich himself (in his 2009 interview with Nelson Hernandez) gives "August or September of 2005" as when he quit his job (and went full-time). Again this error uproots the narrative timeline promoted by Riis. He concludes: "The evidence does not justify an inference that he *must* have copied code." There is ample evidence (see below) that the 2004 and 2005 versions of Rybka are internally unrelated. There was almost certainly, at the least, a dramatic re-write. Furthermore, Riis returns to his *must* standard of evidence, which is inapropos; rather, Rybka 1.0 Beta has elements of substantial similarity with Fruit 2.1, and this suffices.

4.4. **The ICGA process, and a final view.** Riis then has a section making any number of erroneous claims about the ICGA process, some of them defamatory. Others can address this better than I.

Riis concludes with his subjective view, essentially that there was a vendetta against Rybka and its "hypercompetitive genius" leader, Vasik Rajlich. I will let the reader decide if there is any basis behind this hagiography.

In the hopes of reaching a resolution, a goal that endless Rybka Forum posts will certainly not approach, let alone reach, I might propose that Rybka/Rajlich agree to a binding (this is a key aspect) review by an uninterested party. I would suggest Ken Thompson, but he has already weighed in. Similarly, Jonathan Schaeffer should suffice, but perhaps they would fret that he has old-school ICGA connections. It should be possible to procure a suitable panel from eminent programming doyens, though this is likely no longer *gratis*, and the loser should be yoked with the associated costs – thus I doubt any such resolution will ever occur.

5. A MOST RELEVANT ISSUE OMITTED BY RIIS

Riis completely omits the fact that Rajlich had previously passed off the work of others as his own. This was actually a major discovery that was made during the ICGA Panel investigation, made possible by Olivier Deville submitting the relevant Rybka executables to the ICGA.

To wit, in April 2004, and then later in November 2004, Rajlich submitted various private Rybka versions (numbered 1.4 to 1.6) to Deville for entry into the latter's author-based privately-run ChessWar tournaments. The ICGA investigation found that these Rybka versions contained substantial amounts of Crafty code, both in search and evaluation, including a function copied directly into Rybka that runs 100+ lines in Crafty. I very briefly go through some of this evidence in pages 7-14 of my recapitulation PDF linked above.

Besides being copyright infringement of the relevant version of Crafty (19), it appears that Rajlich's actions to gain entry into Deville's tournament were actually fraudulent (based on the 9-point test of common law for this).

Furthermore, there is evidence that Rajlich knew that his actions were inappropriate, but continued them anyway. During mid-2004 another Crafty derivative was discovered, namely ElChinito. Rajlich commented in the contemporaneous CCC thread concerning the issue. These 2004 versions of Rybka were then found to have copied some of the same Crafty bugs that ElChinito did!

5.1. **Jump discontinuty in Rybka internals.** Another highly important aspect of the analysis of these early Rybka versions was that it showed conclusively that the 2004 Rybka versions had essentially very little internal similarity to the 2005 version. For instance, the piece numbering of 2004 Rybka is that of Crafty; for 2005 Rybka it is that of Fruit 2.1 (which differs from Fruit 1.0 incidentally). Another example is that the hashing structure and move encoding were Crafty-based in 2004 Rybka, and suddenly became Fruit-like in 2005 Rybka. Though not of direct chess import, Rajlich also ditched his UCI structure from 2004 Rybka (note that Crafty is not UCI, so this was his own, presumably), adopting a very Fruit-like one with 2005 Rybka (complete with intermixed time management), and omitting previously implemented commands like `searchmoves`.

For the Panel, one result of this evidence was that it cast considerable doubt upon the idea that Rybka had been in "continual development" (except in a rather vague sense) since 2003.

6. CONCLUSION (OR SUMMARY)

In Part I, Riis suspects he has insights that the Panel ignored, and tries to argue that Rule #2 has become inapplicable. Neither of these assertions is borne out by his arguments in the later parts.

In Part II, among other problems, almost every relevant data point in his graph appears to be erroneous. Nothing in this first section holds water, in particular his intended conclusion that Rule #2 is no longer workable. On plagiarism, he asserts that Rajlich's failure to cite Fruit in his ICGA entry is their problem (not his), and then returns to railing against Rule #2. He greatly minimizes the magnitude of Fruit/Rybka overlap here, perhaps as a corollary of erroneous conclusions made

elsewhere. He then over-values the worth of indirect similarity testing as compared to direct code analysis. He finally mis-dates a quotation from Letouzey on originality, thus applying it to the wrong Fruit version.

In Part III, he gives 10 substantial Rybka/Fruit differences in evaluation, of which at most 4 seem viable, and all were noted by the ICGA. He does not discuss the remaining 20-30 evaluation elements, which were largely found to be substantially similar. He then misconstrues the "algorithmic" nature of the evaluation function, ignoring any creative aspects in its creation. He then monstrously exaggerates the impact of the floating-point zero issue on the ICGA decision, doing similarly with the PST issue in the next part. He fails to address a number of additional Fruit/Rybka congruences that were detailed by the investigation; together these helped to form a much broader picture than the 2 or 3 elements that he presents.

In Part IV, he misinterprets the question of PST copying. He seems to invent his own measure of comparison (raw numbers), while a proper metric would be the number of Fruit code changes necessary to replicate the Rybka output (and whether this number was abnormal). He then eccentrically proposes that one can skirt any "copying" issue by translating to a different programming language. He then suggests a timeline for Rybka development that erroneously states Rajlich went full-time in 2003, rather than in mid-2005. I will not address his defamation of Hyatt and the ICGA.

Riis omits any mention of the fact that Rajlich had previously plagiarized Crafty in private 2004 versions of Rybka, and furthermore that these versions had little internal similarity to the 2005 Rybka. The latter fact played a significant role in the Panel deliberations, strongly implying 2005 Rybka was a re-write, at the least.

Throughout, Riis displays little knowledge of programming, let alone that of computer chess therein. For instance, he claims that a one-time operation that takes less than a millisecond on Fruit startup (or during compile with Rybka) should be made fast for efficiency reasons. Furthermore, he is similarly lacking in any knowledge of the relevant aspects of copyright law, particularly the Abstraction-Filtration-Comparison Test that formed a basis for one part of the ICGA Panel analysis. He appears to apply a minimalist copy/paste standard to what "copying" might mean, ignoring any other creative aspects. Finally, he consistently refuses to apply any inferential capability regarding likely scenarios; combining this with an artificially impossible standard of proof, he is reduced to the pedantry of repeatedly asserting that no one can *prove* that Rajlich directly copy/pasted Fruit source code, when this was never the issue to start.