Was Stellungstests testen

Ich habe einen Stellungstest kreiert. Und weil Bescheidenheit noch nie meine Schwäche war, habe ich ihn *Lars-Bremer-Test* genannt. Fairerweise müsste er *Lars-Bremer-Test based on WM-Test* heißen, aber das war mir zu lang. Das Besondere am Lars-Bremer-Test ist seine bemerkenswerte Flexibilität – es erringt nämlich immer diejenige Engine den Spitzenplatz, von der ich möchte, dass sie gewinnt. Dabei enthält der Lars-Bremer-Test mehr Stellungen als der WM-Test Express, nämlich mindestens 40. Ausgewogen ist er auch, denn das Verhältnis der Aufgaben aus den Kategorien Endspiel, Königsangriff und Positionsspiel ähnelt dem des Original-WM-Tests, falls der Anwender das wünscht. Dass es sich ausschließlich um hochwertige und sorgfältig ausgewählte Aufgaben handelt, dürfte ohnehin jedem bekannt sein.

Test-Autor Walter Eigenmann aus der Schweiz fragte vor einiger Zeit, ob ich für ihn nicht ein kleines Programm schreiben könne. Die Fritz-GUI löst Testsuites automatisch und schreibt die Ergebnisse der getesteten Engines als Kommentar in die Datenbank der Teststellungen. Walter Eigenmann wollte ein Tool, das diese Datenbank einlesen kann und die Aufgaben in einer Liste darstellt, in der er einzelne Aufgaben aktivieren und deaktivieren wollte. Das Tool sollte eine Rangliste der Engines erstellen und dabei nur die ausgewählten Aufgaben berücksichtigen. Ich habe das Tool geschrieben und festgestellt, dass so ein Prográmmchen ein lustiges Spielzeug ist.

Was Stellungstests testen

Die besten Analysen

Dank Manfred Meilers unermüdlicher Arbeit stehen die WM-Test-Ergebnisse von mehr als 300 Engines, welche alle über den 100-Aufgaben-Parcours gejagt wurden, in einer 1,5 MByte großen PGN-Datei bereit. Beste Engine ist irgendein Fritz. Das finde ich ärgerlich, denn ich mag Shredder viel lieber. Also habe ich die Datei mit meinem Test-Tool eingelesen und alle Aufgaben deaktiviert, die Fritz schneller löst als Shredder. Sowas tut das Tool automatisch, es dauert nur eine Sekunde. Das Ergebnis ist ein Lars-Bremer-Test aus insgesamt 58 Aufgaben, 13 zum Endspiel, 22 Aufgaben zum Königsangriff und 23 zum Positionsspiel. In diesem Test okkupieren die verschiedenen Shredder-Varianten und -Inkarnationen sämtliche Spitzenplätze; hier das Ergebnis von ElostatTS:

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Shredder8_GambitShredd._256MBHash	2770	3	17127	73.5%	2593	55/58
2	Shredder7.04_GambitShredd._256MBHash	2754	3	16842	71.6%	2593	53/58
3	Shredder9UCI_default_256MBHash	2753	3	17151	71.5%	2593	54/58
4	Shredder9UCI_Gambit_256MBHash	2743	3	16124	70.3%	2594	49/58
5	Shredder8_GambitAggr._256MBHash	2742	3	16610	70.2%	2594	52/58
6	Shredder8_default_256MBHash	2738	3	16328	69.7%	2594	51/58
7	Shredder7_GambitShredd._256MBHash	2730	3	15737	68.5%	2595	47/58
8	Fritz8Bilbao_8.1.2.0_256MBHash	2710	3	15327	65.9%	2595	45/58
9	Shredder7.04_Haddad1.2DP5_256MBHash	2706	3	16227	65.6%	2594	50/58
10	Shredder8_Aggressive_256MBHash	2706	3	15861	65.5%	2594	48/58
11	Shredder7.04_default_256MBHash	2702	3	15625	65.0%	2595	47/58
12	Shredder7.04_Comb=Some_256MBHash	2695	3	15495	64.0%	2595	46/58
13	CM10thEdition_Schumacher_128MBHash	2694	3	14906	63.8%	2596	44/58
14	X3DFritz_8.1.1.0_256MBHash	2694	3	14678	63.7%	2596	41/58
15	Gandalf6.0_Gambit_256MBHash	2694	3	15373	63.9%	2595	46/58
16	Shredder7_ElCid_256MBHash	2694	3	15122	63.9%	2595	43/58
17	Shredder6.02_GambitShredd._256MBHash	2692	3	15310	63.6%	2595	44/58
18	Gandalf6.0_Default_256MBHash	2692	3	15725	63.6%	2595	48/58
19	DeepFritz8_8.1.1.0_256MBHash	2691	3	14584	63.4%	2596	41/58
20	Shr9/Gandalf6_TripleBrain_128/104MB	2690	3	15974	63.4%	2594	49/58
21	Shredder7SE_default_256MBHash	2688	3	15233	63.1%	2595	45/58
22	Shredder7_default_256MBHash	2686	3	15284	62.8%	2595	45/58
23	CM9_T05_Hiarcs8-GUI/txt_128MBHash	2683	3	14539	62.2%	2597	42/58

Aber Shredder ist ein sehr starkes Programm, es ist bestimmt kein Wunder, dass man die Engine so leicht an die Spitze bringen kann. Vor ein paar Jahren habe ich in Paderborn die Autoren von Gromit (jetzt Anaconda) kennen gelernt. Frank Schneider und Kai Skibbe sind zwei außergewöhnlich nette Kerle, aber niemand würde wohl behaupten, ihre Engine sei das welt(meister)beste Analyseprogramm. Niemand? Doch, ich! Der Lars-Bremer-Test zeigt ganz klar, dass kein Programm besser analysieren kann als der gute alte Gromit. Dabei enthält der Test immerhin 43 Stellungen (mehr als der Ur-WM-Test!), 7 aus dem Endspiel, 20 Positionsaufgaben und 16 Königsangriff-Tests. Alles "hochwertige und überwiegend korrekte" Aufgaben!

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Gromit3.9.5_CB-native_284MBHash	2734	3	13201	69.0%	2595	43/43
2	Fritz8Bilbao_8.1.2.0_256MBHash	2711	4	11506	65.9%	2597	34/43
3	Ch.Tiger2004_Gambitagress._192MBHash	2702	4	11272	64.6%	2597	33/43
4	CM9_T05_Hiarcs8-GUI/txt_128MBHash	2695	4	11460	63.8%	2597	34/43
5	CM9000-Pillen_TheKing3.23_128MBHash	2694	4	11520	63.7%	2596	34/43

Oder wie wäre es mit dem Chess Tiger? Bei den Schweden liegen die letzten Versionen fast 100 Computer-Elo hinter der Spitze, aber seit ich ansehen musste, wie der Tiger in Ingo Althöfers Wohnzimmer den mit Hiarcs bewaffneten Großmeister Rogozenko wegputzte, halte ich viel von den Tigern. Und der Lars-Bremer-Test mit 43 Aufgaben zeigt es deutlich: Alle Tiger sind einfach spitze!

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Tiger15.0-CP_Gambitaggress._192MBHash	2769	3	13201	73.2%	2594	43/43
2	Ch.Tiger2004_Gambitagress._192MBHash	2767	4	12682	72.9%	2595	40/43
3	Ch.Tiger2004_Gambit_192MBHash	2760	3	12852	72.1%	2595	41/43
4	Tiger15.0_Gambit_192MBHash	2757	4	12402	71.6%	2596	38/43
5	GambitT.2.0_Stil=aggressiv_192MBHash	2752	3	12417	71.1%	2595	39/43
6	GambitT.1.0_default_192MBHash	2743	4	12009	70.1%	2596	37/43
7	GambitT.2.0_Stil=normal_192MBHash	2727	4	11751	68.0%	2596	35/43
8	Ch.Tiger2004_Normal_192MBHash	2705	4	11890	65.2%	2597	35/43
9	DeepJunior9_Single-CPU_244MBHash	2702	4	11474	64.6%	2597	33/43
10	Tiger15.0_Normal_192MBHash	2699	4	11589	64.2%	2598	33/43
11	ProDeo1.0_Q3-TacticalEng._200MBHash	2695	5	10865	63.6%	2598	28/43

Habe ich gerade schlecht über Hiarcs geschrieben? Wie dumm von mir, denn auch Hiarcs beherrscht im Lars-Bremer-Test die Konkurrenz nach Belieben und löst 43 von 44 Aufgaben:

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Hiarcs9_default_256MBHash	2736	3	13303	69.4%	2594	43/44
2	Hiarcs9aggr._Aggressiv_256MBHash	2731	3	13332	68.7%	2594	43/44
3	Gandalf6.0_Gambit_256MBHash	2714	4	12155	66.5%	2595	37/44

Natürlich klappt das Ganze auch mit Ruffian; 11 Endspielaufgaben, 17 zum Positionsspiel und 12 zum Königsangriff sehen die schon leicht angestaubte Engine klar in Front:

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Ruffian2.1.0_UCI-Version_256MBHash	2729	3	12098	68.3%	2595	39/40
2	Ruffian2.0.2_UCI-Version_256MBHash	2725	3	12280	67.9%	2595	40/40
3	Ruffian2.0.0_UCI-Version_256MBHash	2724	3	12280	67.8%	2595	40/40
4	RuffianLeiden_UCI-Version_256MBHash	2712	3	11970	66.2%	2595	38/40
5	Ruffiandevel_v.2003-10-29_256MBHash	2712	3	12075	66.1%	2595	39/40
6	Ruffian06/2003_v.2003-06-23_256MBHash	2709	3	11970	65.8%	2595	38/40
7	Shredder8_GambitShredd._256MBHash	2706	4	11198	65.3%	2596	33/40

Für Chessmaster-Fans habe ich eine gute Nachricht: Praktisch alle Settings haben eine Analysefähigkeit, die weit jenseits der der Konkurrenz liegt, wie ein 48-Aufgaben-Semi-Express-Test beweist:

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	CM9000-Pillen_TheKing3.23_128MBHash	2753	3	14469	71.5%	2594	47/48
2	CM9-Grailm.7_Fritz8-GUI/txt_128MBHash	2740	3	13830	69.7%	2595	43/48
3	CM9000_WHx_Fritz8-GUI/txt_64MBHash	2735	3	14349	69.2%	2594	46/48
4	CM10thEdition_Schumacher_128MBHash	2735	3	13856	69.2%	2595	43/48
5	CM9000R1_Hiarcs8-GUI/txt_192MBHash	2734	3	14114	69.1%	2594	45/48
6	CM9_SKR_Hiarcs8-GUI/txt_128MBHash	2733	3	13729	68.9%	2595	43/48
7	CM9_Mapi_Arena/Wb2Uci_128MBHash	2728	3	13909	68.3%	2595	44/48
8	CM9_T05_Hiarcs8-GUI/txt_128MBHash	2721	3	13769	67.4%	2595	43/48
9	CM9_Gladiator_Hiarcs8-GUI/txt_128MBHash	2719	3	13820	67.1%	2595	43/48

10	CM8000-Pillen_TheKing3.12d_256MBHash	2714	3	13658	66.5%	2595	43/48
11	CM9_Kleinert_CM9000-GUI_128MBHash	2714	3	13934	66.5%	2595	44/48
12	CM8000-Xpv3_TheKing3.12d_128MBHash	2713	3	13470	66.4%	2595	42/48
13	CM9000_M2v5_Fritz8-GUI/txt_128MBHash	2712	3	13789	66.3%	2595	43/48
14	CM8000-Pillen_TheKing3.12d_32MBHash	2709	3	13325	65.8%	2596	41/48
15	CM10thEdition_CM10thEdR1X_128MBHash	2708	3	13536	65.7%	2595	41/48
16	Fritz8Bilbao_8.1.2.0_256MBHash	2696	3	12884	64.0%	2596	38/48
17	CM8999-Brann_TheKing3.12d_128MBHash	2696	3	13227	64.0%	2596	40/48
18	CM8000-Schüle_TheKing3.12d_256MBHash	2694	3	13048	63.7%	2596	39/48
19	Gandalf6.0_Gambit_256MBHash	2691	3	13139	63.4%	2596	40/48
20	DJunior8.ZX_Single-CPU_255MBHash	2690	4	12607	63.0%	2597	36/48
21	CM9_Utz12n_TheKing3.23_128MBHash	2688	3	13950	63.1%	2595	43/48
22	Gandalf6.0_Default_256MBHash	2688	3	13093	63.0%	2596	40/48
23	CM8000-GS_TheKing3.12d_128MBHash	2685	3	13294	62.5%	2596	41/48
24	CM9000_TheKing3.23_128MBHash	2684	3	13370	62.5%	2596	40/48
25	DeepJunior9_Single-CPU_244MBHash	2684	4	12900	62.2%	2597	36/48
26	CM-Grailmast.2_TheKing3.12d_128MBHash	2679	3	13128	61.7%	2596	40/48
27	CM10thEdition_Default_128MBHash	2678	3	13317	61.6%	2596	40/48
28	CM9_Utz12p_Hiarcs8-GUI/txt_128MBHash	2678	3	13157	61.5%	2596	39/48
29	Shredder8_GambitShredd._256MBHash	2677	4	12592	61.4%	2597	36/48
30	CM8777-Kleinert_TheKing3.12d_64MBHash	2672	3	13266	60.7%	2596	40/48

Was habe ich gemacht? Nur eine Auswahl getroffen; Stellungen verwendet, die den jeweils an die Spitze zu bringenden Engines liegen, die anderen weggelassen. Die Rangliste enthielt in jedem einzelnen Beispiel sämtliche getesteten Programme; ich habe nur Aufgaben, keine Engines weggelassen. Es klappt mit allen Programmen aus dem oberen Drittel, sie alle können die Pole Position in meinem Test erobern. Und selbst Oldies wie Genius lassen sich locker in die Top-Ten hieven. In ausgewogenen Tests, wohlgemerkt, mit Aufgaben aus allen drei Bereichen, und mehr Aufgaben, als der Express-WM-Test enthält. Und sage niemand, die Stellungsauswahl müsse so sein, dass die beste Engine nicht alle Aufgaben löse; es wäre gar kein Problem, noch ein paar Stellungen dazuzunehmen, welche die Rangliste an der Spitze kaum verändern, weil keine der oben postierten Engines sie lösen, sondern nur Programme aus dem mittleren Bereich der Liste.

Was Stellungstests testen

Der Quatsch-Test

Der Lars-Bremer-Test ist in allen Variationen ein Quatsch-Test; was soll er schon aussagen, wenn eine vorher bestimmte Engine absichtlich an die Spitze gebracht wurde? Aber er ist ausgewogen im Sinne des WM-Tests, enthält also Aufgaben aus dem Endspiel, dem Königsangriff und dem Positionsspiel. Zudem besteht er ausschließlich aus WM-Test-Aufgaben. Wenn man ein paar Lars-Bremer-Quatsch-Tests kombiniert, entsteht wieder der komplette WM-Test. Aber was kommt heraus, wenn man aus einer Hand voll offensichtlich blödsinniger Quatsch-Tests einen großen Test bastelt? Entsteht ein vernünftiger, fairer und aussagekräftiger Gesamt-Test oder nur ein etwas größerer Quatsch-Test, bei dem der Quatsch nur nicht mehr so leicht zu erkennen ist, weil er einmal kräftig umgerührt wurde?

Kann der WM-Test wenigstens eine grobe Reihung der Engines vornehmen, nach welchen Kriterien auch immer? Kann er nicht, denn wie gezeigt kann man sehr leicht für jede beliebige Engine des oberen Drittels und für etliche aus dem mittleren Drittel ein Subset von mehr als 40 Prozent der Aufgaben finden, in denen diese Engine die beste ist! Wobei noch zu erwähnen wäre, dass ich mir keine besondere Mühe gegeben und pro Auswahl nur etwa zwei Minuten investiert habe; es ließen sich für jede der an die Spitze geholten Engines auch locker mehr als 50 Aufgaben finden, wenn man sich fünf oder mehr Minuten Zeit für die Auswahl nähme. Was aber die Mühe kaum lohnen würde.

Der WM-Test eignet sich also nicht einmal, um mit Sicherheit festzustellen, ob eine Engine ins obere oder ins mittlere Drittel einer Liste gehört; und je weiter unten eine Engine steht, je weniger Stellungen sie also löst, desto zufälliger wird das Ergebnis. Bescheidet man sich mit 30 Aufgaben wie der WM-Test Express, kann man praktisch jede Engine außer ein paar ganz schwachen Amateuren weit nach oben bringen. Zum Beispiel den Oldie MChess, der hier nur wenige Pünktchen hinter Fritz und Junior liegt:

	Program	Elo	+/-	Matches	Score	Av.Op.	S.Pos.
1	Fritz8 Bilbao_8.1.2.0_256MBHash	2709	4	8039	65.4%	2598	24/30
2	DeepJunior9_Single-CPU_244MBHash	2706	5	8515	65.1%	2598	25/30
3	MChessPro8_default_60MBHash	2701	4	9003	64.5%	2597	29/30

Wie wenig ein Express-Test mit nur 30 Stellungen aussagt, kann man auch daran sehen, dass ein ausgewogener Express-Quatsch-Test mit ebenfalls 30 Stellungen den WM-Test-Spitzenreiter *Fritz Bilbao* erst auf Platz 244 von 308 getesteten Engines einreiht!

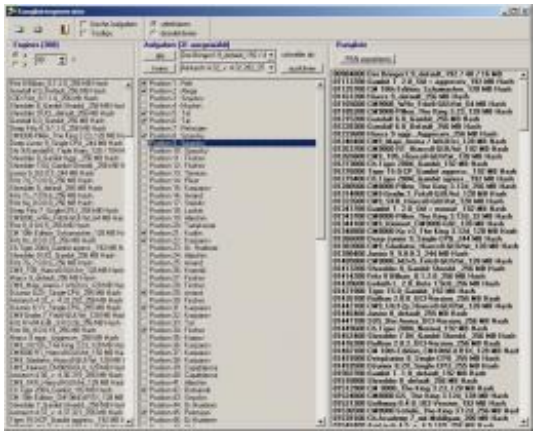
Um in einem in akzeptabler Zeit durchführbaren Test keiner Engine absichtlich einen Vorteil zu verschaffen, müsste man aus einer extrem großen Menge Teststellungen eine Untermenge *zufällig* auswählen lassen. Was bedeutet das für das Testergebnis? Klar, es wäre ebenfalls *zufällig*. Es könnte *zufällig* eine Engine bevorteilen und eine andere benachteiligen. Um statistisch halbwegs auf der sicheren Seite zu sein, bräuchte man sehr viele Aufgaben, weit mehr als 100. So richtig vermag aber auch dies nicht zu verwundern; etliche Schachliebhaber haben sich angesichts der unerschöpflichen Vielfalt unseres Spiels schon lange gefragt, wie man mit 100 (oder noch weniger!) Stellungen Aussagen über die allgemeine Analysefähigkeit einer Engine treffen will.

An dieser Stelle kommt eine andere Schwäche des Lars-Bremer-Tests ins Spiel, die er neben seiner Willkürlichkeit und Zufälligkeit vom WM-Test geerbt hat: Es fällt auf, dass die Gambit-Settings verschiedener Programme oft besser abschneiden als die Normaleinstellungen, welche die Autoren für besser halten. Ein ziemlich bekannter Programmierer meint, das liege daran, daß der WM-Test vorwiegend Stellungen enthält, in denen ein brillanter Zug möglich ist. Das schreiben die Autoren des WM-Tests auch selbst in CSS 5/2001: "Jeder zu suchende Lösungszug stellt einen menschlichen Geniestreich ... dar." Solche "Geniestreiche" finden Spezial-Settings mit besonderem Gewicht auf Königssicherheit, hoher Freibauernbewertung usw. oft schneller; aussagekräftiger wäre nach Ansicht des Programmierers ein Test, der auch Stellungen enthält, in denen ein blendender Zug scheinbar zwar möglich sei, aber eben nicht die beste Möglichkeit darstelle.

Was Stellungstests testen

Zufall oder Methode?

Die Auswahl der Teststellungen entscheidet ganz offenbar über die Reihung der Engines in der Test-Rangliste; eigentlich eine Binsenweisheit, aber sie demonstriert das Dilemma eines Testautors: Wie wählt er aus der Vielzahl der Schachstellungen die aus, die letztlich im Test enthalten sein sollen? In der Einleitung des allerersten WM-Test-Artikels (CSS O5/2001) steht, die Autoren hätten "unzählige Turniere und Matches untersucht". Testerfinder Michael Gurevich hat oft erwähnt, die Stellungen dürften weder zu leicht noch zu schwierig sein. Das ist einzusehen, denn ein Test, dessen Aufgaben kein Programm schafft, wäre ja ganz nutzlos, ebenso wie einer, dessen Aufgaben für kein Programm eine Herausforderung darstellen. Um das feststellen zu können, muss ein Testautor also alle Aufgaben *intensiv mit mehreren Programmen untersuchen*. Das ist auch noch aus einem anderen Grund erforderlich, denn wie mir ein anderer Testautor mitteilte, helfen bei der Überprüfung auf Korrektheit einer Aufgabe die Kommentare der Großmeister nur sehr eingeschränkt. Ein Testautor wählt also unter einer Reihe von Stellungen, *die er alle mit vielen verschiedenen Engines untersucht hat*, die aus seiner Sicht geeignetsten für den Test.



Das Testtool erzeugt Wunsch-Ranglisten

Damit schwimmt ein wichtiges Prinzip wissenschaftlicher Arbeit den Bach hinunter, denn die Auswahl der Stellungen erfolgt mithilfe der Engines, die mithilfe der Stellungen eingeschätzt werden sollen, die mithilfe der Engines ausgewählt wurden, die mithilfe der Stellungen ... Kurz: Was ein Test eigentlich erst ermitteln soll, wird verwendet, um den Test selbst zu erstellen und zu überprüfen. Die Autoren werden wohl darauf achten, dass die Stellungen nicht alle von derselben Engine gelöst werden, sondern immer von anderen ... halt, hatten wir das nicht schon? Klar, es ist genau dasselbe, was ich mit der Kombination der verschiedenen Lars-Bremer-Tests gemacht habe, als sie zum WM-Test verschmolzen.

Es gibt noch eine andere Möglichkeit. Wenn ein Autor wünscht, sein Test möge eine Rangliste abbilden, etwa die der SSDF oder die CSS-Rangliste, könnte er so lange mit verschiedenen Aufgaben-Kombinationen herumprobieren, bis das Testergebnis einigermaßen mit den Ranglisten übereinstimmt. Doch welche Schlußfolgerungen soll man aus einem Test ziehen, der genau auf die Ergebnisse geeicht ist, die er doch eigentlich erst ermitteln soll?

Wie auch immer der Test zustande kam, er sagt nichts aus über das Verhalten von Engines, die es zum Zeitpunkt seiner Erstellung noch nicht gab. Neue Engine-Versionen verhalten sich vielleicht halbwegs vernünftig, weichen also nicht zu sehr vom erhofften Ergebnis ab; manchmal tun sie es aber doch, was dann natürlich an den Engines liegen soll und nicht am Test! Sobald aber eine völlig neue Engine auf dem Mark erscheint, die es bei der Testerstellung noch nicht gab, muss ein Test für diese zufällige Ergebnisse liefern. Fruit ist ein gutes Beispiel; scheinbar ein Versager in Teststellungen liegt das gemessen an der praktischen Spielstärke schlechte Abschneiden im WM-Test einzig daran, dass die Engine eben noch nicht existierte, als die Teststellungen ausgewählt wurden, nicht etwa an schlechter "Analysefähigkeit". Zufällig schneidet Fruit schlecht ab; es hätte auch umgekehrt kommen können, aber auch das wäre nur zufällig geschehen.

Fazit

Der WM-Test hat ein bisschen Pech. Ich habe meine Experimente damit gemacht, weil er der umfangreichste und verbreitetste Test ist und weil für sehr viele Engines Ergebnisse vorliegen. Es hätte wahrscheinlich auch mit jedem anderen Stellungstest funktioniert, denn ich halte die Schwächen der Tests für systemimmanent. Dabei habe ich gar nichts gegen Stellungstests und finde ganz im Gegenteil, dass sie eine sehr angenehme Art darstellen, das Hobby Computerschach mit "richtigem" Schach zu kombinieren. Außerdem eignen sie sich sehr gut, um bestimmte für typisch gehaltene Stärken oder Schwächen einer Engine zu veranschaulichen; nicht als Stellungs*test*, sondern als Sammlung von Teststellungen. Stellungstests eignen sich aber nicht, um aus den Ergebnissen Ranglisten zu erstellen, zumindest nicht, falls man zufällig aus diesen Listen irgendwelche Schlüsse ziehen möchte.

Die Ergebnisse des betrachteten WM-Tests zeigen ausschließlich, wie die Engines mit diesen speziellen Teststellungen zurechtkommen. Praktisch jede beliebige andere Auswahl an Teststellungen führt zu anderen Resultaten. Es gibt keine Korrelation mit Ranglisten, die auf Partien basieren, außer einer unabsichtlich oder willkürlich *herbeigeführten* Korrelation und zufälligen Übereinstimmungen. Auch über die so genannte Analysefähigkeit geben Tests angesichts der gewaltigen Ergebnis-Schwankungen nur höchst eingeschränkt Auskunft. Wer mehr als "könnte vielleicht ganz brauchbar sein" oder "ist wahrscheinlich nicht der beste Tipp" über eine Engine aus den Ergebnissen eines Stellungstests herausliest, lügt sich selbst in die Tasche. Ganz klar, man wird immer bestimmte Stellungen finden, deren Auswahl eine Engine bevorzugt. Aber wenn es so viele sind, wenn man mit der Hälfte der Teststellungen einen beliebigen Spitzenreiter küren kann, wenn man in einem Express-Test den Spitzenreiter des Gesamttests bis ins hintere Drittel durchreichen kann, dann verdienen die Ergebnisse dieses Tests nur das Prädikat *zufällig*.

Der oben erwähnte Walter Eigenmann, für den das Testtool ja bestimmt war, findet meine Sichtweise natürlich völlig falsch. Als ich ihm diesen Artikel zu lesen gab, bot ich ihm gleichzeitig an, in einer der nächsten Ausgaben von CSS Online seine Sicht der Dinge darzulegen. Mit den Worten: "Es wird mir dann ein ganz besonderes Vergnügen sein, deinen Artikel zur Makulatur zu machen!" nahm er das Angebot schmunzelnd an. Es bleibt also spannend! *(Lars Bremer)*