

Lars Bremer

Das Rückwärts-Genie

Endspielanalyse mit Yace

Kein Programm eignet sich so gut für Analysen wie Yace. Dabei entdeckt das Programm viel versprechende Fortsetzungen keineswegs eher als andere, und auch in diversen Stellungstests ist es keine Kanone. Was Yace für Analysefreunde so geeignet macht: Es merkt sich einmal berechnete Bewertungen besser als jedes andere Programm.

Kennen Sie den Tante-Frieda-Benchmark? Damit misst man die „Briefeschreib-Tauglichkeit“ eines Computers – aber inklusive Hochfahren. Also die Zeit zwischen Rechner anknipsen, Betriebssystem starten, Textverarbeitung starten, einen kurzen Brief an Tante Frieda schreiben und diesen drucken. Erst wenn der Brief fertig aus dem Drucker rutscht, stoppt man die Zeit. Zugegeben, das ist kein ganz ernsthafter Test, aber sagt ein synthetischer Benchmark, der etwa misst, wie viele Millionen Textzeilen der Rechner pro Sekunde formatieren kann, wirklich mehr über die Eignung des Geräts als Schreibmaschine?

Beim Schach ist es ganz ähnlich – es kursieren verschiedene Stellungstests, die mehr oder weniger zutreffende Aussagen darüber versprechen, wie gut ein Programm denn zur Analyse von Schachstellungen geeignet ist. Andere Tester lassen viele Partien spielen, um aus den entstehenden Ranglisten Rückschlüsse zu ziehen. Erprobte Verfahren, aber warum eigentlich *analysiert* niemand mit einer Engine, um ihre Analysestärke zu ermitteln? Schnell bekommt man mit, dass es ziemlich unerheblich ist, ob ein Programm einen Lösungszug nach fünf oder nach zwanzig Minuten findet – so lange wartet ohnehin niemand. Und es stellt sich heraus, dass eine Eigenschaft sehr wichtig ist, die kaum jemand im Zusammenhang mit Analysefähigkeit erwähnt: wie sich die Engine einmal berechnete Varianten merkt.

In Stein gemeißelt

Früher, als Computerschach noch etwas für richtige Männer war, haben die Programme ihre Hashtabelle, in der sie sich schon berechnete Stellungen merken, nach jedem Zug gelöscht. Das war nötig, weil sie die auf dem Brett stehende Position sehr

gründlich untersuchten und ihre Bewertungsfunktion modifizierten, bevor sie mit der Suche begannen. Weil die Bewertung abhängig von der Wurzel-Stellung war, durften Hash-Einträge nicht länger als einen Halbzug lang leben, weil sonst die gespeicherte Bewertung nicht gestimmt hätte. Diese *Prescan* genannte Technik hat auf den langsamen Computern anno dazumal sehr gut funktioniert, denn die erreichbare Suchtiefe war ziemlich gering und die Stellungen, die bewertet werden mussten, nie allzu weit von der aktuellen Brett-Position entfernt.

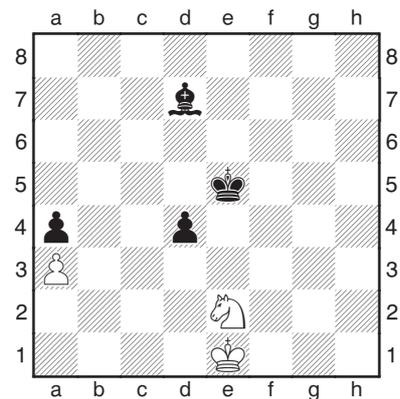
Die Rechner wurden schneller, die Suchtiefe größer, und das *Prescan*-Konzept stieß an seine Grenzen; zu oft hatte die bewertete Stellung nichts mehr mit der Brettstellung zu tun, die vorab ausgeführten Änderungen der Bewertung erwiesen sich immer öfter als kontraproduktiv. *Prescan* flog über Bord, und plötzlich war es gar nicht mehr nötig, nach jedem Zug die Hashtabellen zu löschen, weil die Bewertung nicht mehr modifiziert wurde. Das Programm konnte auf im vorigen Zug berechnete Hash-Einträge zurückgreifen und, noch bedeutsamer, bei einer Rückwärts-Analyse auf wertvolle Hash-Einträge aus einer späteren Partiephase.

Die persistente Hashtabelle war geboren. Das erste kommerzielle Programm mit dieser Funktion, *Hiarcs 7.32*, schlug ein wie eine Bombe. Eine Stellung zu analysieren ist schließlich viel leichter, wenn das Programm dazulernt und nicht jedes Mal denselben Blödsinn berechnet. Man konnte *Hiarcs*, wenn es eine Stellung falsch bewertete, die Widerlegung zeigen und das Programm bewertete dann auch die Ausgangsstellung anders und womöglich genauer.

Heute speichern alle Programme ihre Berechnungen in dauerhaften Hashtabellen. Alles wäre gut, wenn

diese unendlich groß wären. Sind sie aber nicht, und so steht eine Engine oft vor der Entscheidung, ob es eine gespeicherte Stellung mit einer neu berechneten überschreiben soll oder nicht. Es gibt verschiedene Entscheidungskriterien dafür und jedes Programm verwendet unterschiedliche Algorithmen; einen Königsweg gibt es leider nicht. Dies führt dazu, dass die Engines sich während der Analyse einer Stellung, wenn der Bediener verschiedene Varianten durchgeht, unterschiedlich gut merken, was schon berechnet wurde.

Und hier schlägt die Stunde des Freeware-Programms *Yace* von Dieter Bürßner, das die kommerziellen Programme wie *Fritz*, *Shredder* und auch *Hiarcs* gnadenlos abhängt, wenn es um Rückwärtsanalyse geht. Um die Fähigkeiten von *Yace*, aktuell ist Version 0.99.87, im Vergleich zu Spitzenprogrammen zu testen, haben wir eine Stellung aufgebaut, die in einer Partie zwischen *Shredder* und *Yace* im Paderborner Turnier 2003 hätte entstehen können:



Schwarz am Zug

Ziel des Experimentes war herauszufinden, ob wir ein Programm dazu bringen können, für diese Stellung eine *Mattbewertung* anzuzeigen. Dazu war es nötig, den Engines die Varianten so lange vorzuspielen, bis sie ein Matt erkennen, dann zurückzugehen und dasselbe mit allen anderen Varianten ebenfalls zu tun.

Shredder 8, *Fritz 8*, *Hiarcs 9* und *Yace 0.99.86* (bis auf den verwendeten Compiler identisch mit Version 0.99.87) bekamen jeweils 128 MByte Hash zugeteilt und wurden auf einem Athlon 1400+ ins Variantendickicht geschickt, nur assistiert von den 5-Steiner-Tablebases. Am schlechtesten schnitt *Fritz* ab; nach nicht einmal einer halben Stunde Analyse hatte er wichtige Informationen vergessen.

Hiarcs merkte sich einmal Berechnetes besser, stellte aber nach einer Stunde die Analysearbeit aus unbekanntem Grund komplett ein – die ganze Mühe für die Katz. Am besten von den Spitzenprogrammen zog sich noch Shredder aus der Affäre, der erst nach anderthalb Stunden Entdeckungsreise im Variantenschlingel unter Amnesie zu leiden begann. Aber auch Shredder konnte nicht einmal ein Drittel der komplizierten Abspiele untersuchen, bevor der SSDF-Spitzenreiter wichtige Varianten vergaß.

Yace zeigte erste Merkschwächen nach knapp drei Stunden, die aber nur sehr lokale Geplänkel betrafen, nicht aber wichtige Varianten, in denen viel Arbeit steckt. Nach über fünf Stunden Daueranalyse mit qualmender Maus fing dann auch Yace an zu schwächeln und merkte sich Züge und Bewertungen nicht mehr ganz so gut – ohne jedoch, wie die Profi-Programme, ganz umzufallen. Nach mehr als sechs Stunden und über 1000 analysierten Zügen hatte Yace es geschafft und zeigte Lb5 mit Matt in 71 an. Das ist natürlich nur eine obere Schranke, gut möglich und sogar wahrscheinlich, dass Schwarz dem Weißen auch in weniger Zügen das Lebenslicht ausblasen kann.

Dieses Beispiel mag nicht unbedingt aus dem Leben gegriffen sein, es zeigt aber sehr gut, wo die Stärke des Freeware-Programms im Vergleich zur Profi-Konkurrenz liegt. Auch bei ganz gewöhnlichen Analysen ist Yace mehrere Nasenlängen vor allen anderen.

Tipps und Tricks

Beim Analysieren möchte Yace die Züge wirklich auf dem Brett sehen; auch selbstverständliche Züge sollte man kurz ausführen, die Engine merkt sich das Ergebnis dann besser. Auch Fail-high- und Fail-low-Züge erfordern Sorgfalt: Bei einem fail high, in Fritz durch eine grüne Hauptvariante gekennzeichnet, erkennt die Engine, dass die Seite am Zug besser steht als erwartet. Wie viel besser, weiß sie vorerst nicht. Eine rote Variante markiert ein fail low; die Engine erkennt ein Unglück für die Seite am Zug. In beiden Fällen, besonders aber beim fail low, muss der ungeduldige Analysator warten und darf nicht gleich zum vorigen Zug zurückgehen. Manchmal kann es aber Zeit sparen, die ersten zwei oder drei Züge der an-



Dieter Bürßner, 39, ledig und kinderlos, ist Chemiker von Beruf. Zu seinen Hobbys gehört neben Schachprogrammierung der Radsport. An Yace arbeitet er schon seit etwa zehn Jahren, mit längeren Unterbrechungen. „Angefangen hat alles auf Atari-ST und PC, damals auch auf VAX, MIPS und mehr, was halt gerade an der Uni zur Verfügung stand.“

CSS: Dieter, warum hast du Yace keinen Multivarianten-Modus eingebaut?

DB: Zu faul, zu mühsam, keine Zeit. Es gibt eine gute Alternative, die in Yace eingebaut ist: Man kann den besten Zug von der Suche ausschließen (oder mehrere beste Züge). Sollte oft gleichwertig oder besser sein als der reine Multivarianten-Modus, weil man in gleicher Zeit tiefer kommt. Das geht als UCI-Engine oder im Konsolen-Modus. Klar, es ist nicht so bequem zu bedienen. Wenn ich mal Zeit habe ...



CSS: Wie stellst du das an, dass Yace für Rückwärtsanalysen so immens gut geeignet ist, besser als jedes andere Programm?

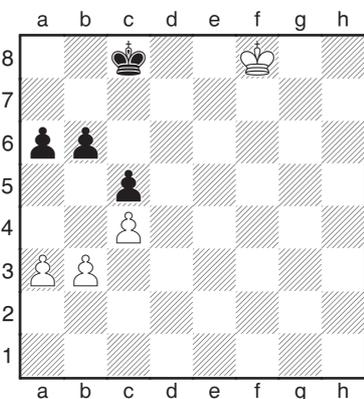
DB: Betriebsgeheimnis. Ich hatte das übrigens schon lange vor. Dann

kam noch eine kleine Motivations-spritze: Uri Blass schrieb im CCC-Forum so ungefähr, Programme könnten das nicht gescheit. Es geht zwar, je nach Programm, eine Weile gut, aber dann vergessen sie wieder, insbesondere wenn man mehrere Varianten betrachten muss. Dann setzte ich mich mal hin und versuchte das zuverlässig zu machen.

CSS: Offenbar interessierst du dich besonders für das Endspiel, Yace weiß eine Menge über diese Partiephase. Was findest du daran spannend?

DB: Vieles ist im Endspiel leichter, etwa Regeln mit weniger Ausnahmen zu finden. Besonderen Wert lege ich auch auf Verlässlichkeit – die Engine soll nie theoretisch versagen können, wie das bei normalem Nullmove leicht passieren kann – auch ewig viel Zeit kann dann nicht helfen. Yace sollte theoretisch, also mit unendlich viel Zeit, keine solche Schwäche haben.

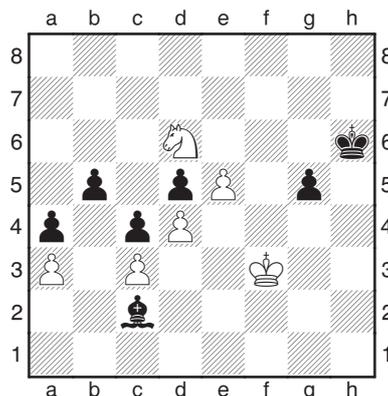
Weiß am Zug gewinnt nur mit Kh2. Unsere Profi-Programme sehen Weiß im Vorteil und wollen Kh2 spielen, bewerten aber Kg1 und Kh2 fast gleich. Das ändert sich auch nach einiger Wartezeit nicht – weder Shredder, Fritz noch HiarcS verstehen diese Stellung. Yace hält von Anfang an Kh2 für den mit Abstand spannendsten Zug und zeigt mit jeder Iteration eine höhere Bewertung dafür. Nach kurzer Zeit erkennt es dann den endgültigen Gewinn, ganz ohne Rückwärtsanalyse. Stellt man die Tablebase-Nutzung in den Engine-Parametern auf „aggressiv“, wirft Yace sogar nach einer etwas längeren Weile eine Mattbewertung aus!



Nur a4 gewinnt in dieser Studie von Awerbach, die auch aus einer praktischen Partie stammen könnte.

HiarcS und Fritz versagen hier total und wollen Ke8 mit großem Plus spielen, was wegen eines hübschen Patt-Tricks nicht zum Gewinn reicht. Fritz erkennt das immerhin nach kurzer Zeit und präferiert dann wie Shredder a4 mit einer Bewertung von etwa plus 0,75 Bauerneinheiten. Yace zeigt dagegen nach einer halben Minute den Gewinn für Weiß an und ist nach einer Minute bei plus acht.

Ein letztes Beispiel noch, das zeigt, wie Yace auch in einer praktischen Partie sein Endspielwissen Gewinn bringend einsetzen kann. In der letzten Runde des Paderborner Turniers 2003 kam es zur Begegnung Shredder–Yace. Shredder musste unbedingt gewinnen und stocherte so lange herum, bis es zu folgender Stellung kam:



Hier spielte Yace **84...b4!!** Shredders Bewertung schoss zunächst in die Höhe, aber das Amateurprogramm hatte mehr Durchblick in der Stellung und gewann.

Fazit

Möglicherweise spielt Yace auch in Engine-Matches unheimlich gut, vielleicht aber auch nicht. Wie sich das Programm im harten Match-Alltag schlägt, zeigen die BfF- und die CSS-Rangliste. Wer aber Yace nicht als Spielzeug einsetzen möchte, sondern als Werkzeug, bekommt hier gratis eine besonders endspielstarke Engine mit einer Analysefähigkeit, die Shredder, Fritz und Co. ganz alt aussehen lässt. Doch zu viel Glück erweckt der Götter Neid, und so haben die fiesen Unsterblichen den Yace-Autor Dieter Bürßner veranlasst, uns Anwendern einen Multivariantenmodus vorzuenthalten. Aber die Hoffnung stirbt zuletzt; vielleicht wird ja irgendwann noch einmal ein Yace mit Multivarianten-Modus erscheinen.

Die aktuelle Version *Yace 0.99.87* finden Sie im CSS-Service zu dieser Ausgabe. Im Internet können Sie unter <http://home1.stofanet.dk/moq/> die ca. 1,5 Jahre alte Version *Yace Paderborn 2003* herunterladen.