

Der Fortschritt ist nicht aufzuhalten

Chrilly Donninger über Endspieldatenbanken

Meine Mutter und ihre Schwiegertochter sind sich in einem Punkt einig. Sie sind mit schrecklich altmodischen Ehegatten gestraft. Seit einem Jahr beschweren sich beide, daß es zur Hauptsendezeit im Fernsehen nur noch Schwachsinn gibt und die wenigen guten Filme erst um Mitternacht anfangen. Während sich Vater und Sohn in ihrer ursprünglichen Meinung bestätigt fühlen und vorschlagen, den Flimmerkasten auf dem Dachboden zu deponieren, drängen die beiden Modernisierinnen vehement auf eine technologische Lösung des Problems. Ein Videorecorder muß her.

Evas Begehr ist nicht die einzige Treibkraft des Fortschritts. Fast ebenso wichtig ist die menschliche Faulheit und die Not. Der Nimzowerkstatt-Eröffnungsguru Alex Kure ist schon seit Jahren der Meinung, daß jedes Spitzenprogramm elementare Endspiele können müßte. Mein wirksamer Standardblock war »Alex, Du hast ja recht, aber Ossi Weiner deckt mich derartig mit Aufträgen ein, daß ich einfach nicht dazu komme«. Als sich Ende 1998 die Wege von Ossi Weiner und der Nimzowerkstatt trennten, stand ich ohne diesen Block da. Ich versuchte es noch mit einem Konter »Kommen solche Endspiele in 1000 Jahren mindestens zweimal vor?« Alex war darauf vorbereitet und präsentierte mir eine Reihe von Turnierpartien, in denen Nimzo auf Grund elementarer Ahnungslosigkeit den Gewinnzug nicht fand. Es blieb mir also nichts anderes übrig, als in den sauren Apfel zu beißen.

Als Aufwärmübung beschlossen wir, das besonders einfache Endspiel »König und Dame gegen König und Bauer auf c2« einzubauen. Alex zeichnete mir die kritischen Felder des weißen Königs auf und meinte: »Das müßte genügen«. Ich programmierte diese Felder ein, probierte es aus, und es funktionierte nicht. Ein Programm durchsucht – in einer Turnierpartie – pro Position mindestens 100 Millionen Stellungen. Es findet daher die unglaublichsten Schleichwege, um eine Regel zu umgehen. Im Falle von König, Dame gegen König und Bauer auf c2 ist dieses Schwindeln für das Programm sogar sehr einfach. Steht der weiße König bereits auf den kritischen Feldern (nahe genug bei c2), dann sagt ihm sein Wissen: »Schwarz steht verloren«. Daraufhin versucht die schwarze Suchroutine eine Umwandlung und findet »Gut is gangen, nix is passiert. Jetzt hat auch Schwarz eine Dame«. Diese schwarze Dame geht zwar wieder verloren, aber wenn die schwarze Suche die Umwandlung nahe genug am Suchhorizont ausführt, kann der Verlust über ihn hinausgeschoben werden. Der Trick funktioniert erstaun-

lich lange. Um König, Dame gegen König und Bauer auf c2 zu programmieren, braucht man also auch Wissen um König, Dame gegen König und Dame auf c1. Wir versuchten ein paar Regeln. Diese waren aber nicht in der Lage, Nimzo vom Schwindeln abzuhalten. Unsere Erkenntnis nach 14 Tagen: So geht es nicht, da muß eine Endspiel-Datenbank her.

Doof und Doofer

Im Rahmen der Vorlesung *Computerschach* an der Uni Linz leite ich das Kapitel Endspiel-Datenbank immer folgendermaßen ein: »Wenn jemand ein starker Schachspieler sein möchte, dafür aber zu blöd ist, wird er Computerschach-Programmierer. Reicht's dafür auch nicht, sollte er es mit Endspiel-Datenbanken versuchen«. Manchmal verirrt sich auch ein heller Kopf in die Vorlesung. Der macht mich dann darauf aufmerksam, daß der Datenbank-Pionier Ken Thompson um einiges mehr am Hut hat als der Vortragende. Am Ende des Kapitels muß dieser helle Kopf aber in der Regel zugeben, daß das Programmieren von Endspiel-Datenbanken im Verhältnis zu einem vollen Programm wirklich einfach ist.

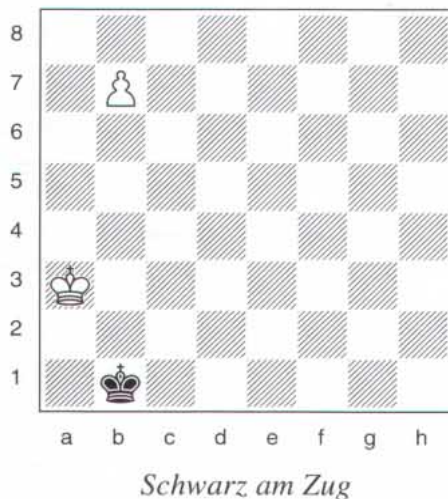
Bei der Programmierung eines bestimmten Endspiels probiert man zunächst alle möglichen Stellungen aus. Es werden alle illegalen Stellungen mit -1 und alle Mattstellungen mit 0 markiert (man kann auch andere Zahlen benutzen). Danach nimmt man aus jeder Mattstellung alle legalen Züge zurück. Man bestimmt also alle Positionen, aus denen mindestens eine Mattstellung in einem Halbzug erreichbar ist, und markiert sie mit 1 (Matt in einem Halbzug). Im nächsten Schritt bestimmt man alle Stellungen, aus denen die Seite am Zug legal nur in eine mit 1 markierte ziehen kann, und kennzeichnet sie mit 2 (Matt in 2 Halbzügen). Aus diesen mit 2 markierten Stellungen nimmt man wieder alle Züge zurück. Falls die Position noch nicht markiert ist (ein Matt in einem Halbzug möglich ist), erhält sie eine 3. Auf diese Art und Weise werden nun sukzessive alle Mattsequenzen berechnet. Das Verfahren bricht ab, wenn in einem Schritt keine neuen Stellungen mehr gefunden werden. Aus noch nicht markierten Stellungen kann man nicht Matt setzen. Sie werden mit einem eigenen Wert für Remis gekennzeichnet. Die Erzeugung einer Endspiel-Datenbank ist also ein retrogrades Verfahren. Man geht von der Lösung (Matt) aus und berechnet rückwärts gehend alle Lösungswege/Mattsequenzen.

Ob eine Stellung matt bzw. illegal ist, hängt neben der Figurenkonstellation auch vom Zugrecht ab. Man muß das Verfahren daher getrennt für jeweils Weiß und Schwarz am Zug durchführen.

Untersucht man z.B. das Endspiel KD-KD (Weiß: König, Dame, Schwarz: König, Dame), dann kann dieses durch das Schlagen einer Dame in KD-K (Weiß: König, Dame, Schwarz: König) übergehen. Das Programm verwendet in diesem Fall das Ergebnis der bereits zuvor erstellten Subdatenbank. Ist die Stellung in KD-K ein Matt in 6, dann addiert man diese Mattdistanz zu den bisher gemachten Rechenschritten und erhält so die Gesamt-Mattdistanz. Die Datenbank KD-KD ist nicht notwendig, da man durch einfache Farbvertauschung das Ergebnis der KD-K-Datenbank verwenden kann.

Dieses Verfahren ermittelt nicht immer die kürzeste Mattdistanz, sondern nur die kürzeste Distanz in eine gewonnene Abtauschstellung. Gemäß der Umwandlungsdistanz spielt Schwarz in der folgenden Stellung am besten Ka1 (verhindert die Umwandlung um einen Zug), das Matt wird jedoch am besten durch Kc2 hinausgezögert. Puristen wie Thomas Mally (Herausgeber der ehemalige Zeitschrift PC-Schach) mögen das unschön finden, aber wegen der 50-Züge-Regel ist die kürzeste Abtauschkosten effizienter.

Teststellung
aus PC-Schach 1/94, S. 61



Thomas Mally stellt in seinem Artikel die Frage, wie Gott spielen würde. Meiner Meinung nach Ka1. Er würde durch diesen Zug dem Menschen eine Chance geben, mit b8=D einen kapitalen Fehler zu machen.

Sieht man von der Feinheit »minimale Mattdistanz versus minimale Umwandlungsdistanz« ab, so enthalten Endspieldatenbanken perfektes Wissen. Selbst ein Matt in 233 läßt sich durch stures Wiederholen einer enttäuschend einfachen Methode berechnen. Es gibt dabei nur ein kleines Problem: Die Größe des Datenfriedhofs. Geht man vereinfachend davon aus, daß jede Figur auf 64 verschiedenen Feldern stehen kann,

dann gibt es bereits für den 4-Steiner KD-KD 64 hoch 4 oder rund 16 Millionen Stellungen. Ein 5-Steiner wie KD-KTS enthält bereits 64 hoch 5 oder rund eine Milliarde Positionen. Diese Zahlen müssen noch mit 2 multipliziert werden, da man die Berechnung jeweils mit Weiß und Schwarz am Zug durchführt. In reinen Figurenendspielen beschränkt man sich allerdings auf Stellungen, in denen der weiße König im Dreieck a1-d1-d4 steht. Durch Spiegelung um die horizontale oder vertikale Mittellinie bzw. um die beiden Diagonalen erhält man alle anderen gleichwertigen Stellungen. Auf diese Weise reduziert man den Aufwand um den Faktor 10/64.

Sind Bauern im Spiel, dann kann man nur um die vertikale Mittellinie spiegeln. Der weiße König wird auf die linke Bretthälfte beschränkt. Besonders giftig sind Endspiele mit Bauern auf beiden Seiten. Durch die notwendige Enpassant-Information erfolgt eine zusätzliche Multiplikation der möglichen Stellungen. Reduziert man diese um Symmetrien, dann bleiben bei den 4-Steinern immerhin noch mehr als 2 bzw. 8 Millionen Stellungen pro Figurenkonstellation und Zugrecht übrig. Insgesamt gibt es 60 verschiedene 4-Steiner. Verwendet man für jede Position ein Byte, dann belegen bereits die 4-Steiner ein paar hundert MByte Festplattenkapazität, alle 5-Steiner wären ein paar hundert GigaByte groß!

Zu Lasten der Hunde

Endspiel-Datenbanken berechnen ist einfach und damit uninteressant. Wesentlich spannender ist das effiziente Abspeichern der Daten. Wie quetsche ich möglichst viele Stellungen auf eine Festplatte bzw. in den Arbeitsspeicher. Endspiel-Datenbanken sind ein Spezialproblem einer kleinen Minderheit. Große Datenmengen möglichst schnell auf möglichst kleinem Raum unterzubringen, ist hingegen das klassische Informatikproblem schlechthin. In der Computer-Steinzeit dachte man, daß es durch verbesserte Hardware und zusätzliche Speicherkapazitäten von selbst verschwinden würde. Einige Jahrzehnte und viele MByte später wissen wir: Je größer die Speichermedien, desto knapper sind sie. Die Informatiker haben sich daher raffinierte Methoden einfallen lassen, um diese Not zu lindern.

Eine besonders elegante Methode zerlegt die Daten in ihr Frequenzspektrum. Diese Umwandlung in Frequenzen funktioniert nicht nur bei Tönen, sondern bei beliebigen Daten (die Mathematik dafür wurde bereits vor 200 Jahren von einem Herrn Fourier geschaffen). Bei der Speicherung läßt man anschließend die hohen Frequenzen, die Obertöne, weg. Die Obertöne benötigen nämlich mit Abstand am meisten Speicherplatz, gleichzeitig tragen sie bei vielen Anwendungen nichts oder nur sehr wenig zur Qualität der Daten bei. So kann man etwa bei Musikaufnahmen Obertöne weglassen, die jenseits der Hörschwelle des menschlichen

TableBases für Zuhause

Die Zahl der Programme, die mit den Fünfsteiner-Endspieldatenbanken von Nalimov umgehen können, wächst ständig. Zur Zeit nutzen neben Nimzo7.32 u.a. Fritz6, Hiarc7.32 und Shredder4 die großen Tabellen. Auf den Installations-CDs der genannten Programme werden meist einige wenige Tabellen gleich mitgeliefert. Für alle 4- und 5-Steiner benötigt man aber fast 8 GigaByte Speicherplatz, dafür reicht die Kapazität einer CD bei weitem nicht aus.

Wer mehr möchte, hat jetzt zwei Möglichkeiten: Unter der Bezeichnung »Fritz-Endspielturbo« gibt es bei ChessBase die wichtigsten Fünfsteiner (ca. 3 GByte) auf vier CDs zum Preis von 98,- DM. Wer alle 290 Endspiele haben möchte, kann beim CSS-Shop (Tel. 040-83293133) die sog. *TableDisk* bestellen, eine betriebsfertige und geprüfte Festplatte, auf der sämtliche Tablebases vollständig enthalten sind. Eine Einbau- und Konfigurationsanleitung ist im Preis von 399,- DM enthalten.

Ohres liegen. Möglicherweise empfinden dies aber Hunde als Katzenmusik. Bei Bildern oder Videos erzeugen einzelne andersfarbige Punkte in einer einheitlichen Fläche oder auch scharfe Konturen diese (mathematischen) Obertöne. Filtert man die Obertöne weg, so bedeutet dies, daß die Fläche einheitlich gefärbt bzw. die Konturen etwas geglättet werden. Innerhalb gewisser Grenzen führt dies ebenfalls zu keiner merklichen Verschlechterung der Bildqualität. Im Extremfall speichert man nur die Grundfrequenz (die mittlere Bildfarbe) ab. Ein ganzes Bild könnte auf diese Weise auf ein paar Bytes reduziert werden. Der russische Maler Malewitsch hat auf diese Weise bereits 1917 eine Winterlandschaft auf einer Leinwand abgespeichert. Er nannte es »Das weiße Quadrat« und wurde berühmt dafür. Moderne PC-Benutzer dürften diese Radikal-Methode aber eher als Reduktion der Bildqualität empfinden.

Schachschwingungen

Das Frequenzverfahren könnte man auch für Schachdatenbanken anwenden. Die Komprimierung wäre beeindruckend, das Ergebnis aber leider nicht. Im Gegensatz zu Video- oder Audioaufnahmen hat man noch keine sinnvolle Interpretation für hohe Frequenzen gefunden. Die Schachdatenbank würde einfach viele falsche Bewertungen aufweisen. Fehler in 4-Steinern würden sich in 5-Steinern wie die Kaninchen fortpflanzen. Für die Komprimierung einer Endspiel-Datenbank muß man daher auf die weniger effektiven verlustfreien Verfahren zurückgreifen. Bei einem solchen Verfahren erhält man nach dem Entpacken wieder die exakten Ausgangswerte.

Ken Thompson konstruierte einen erstaunlich gut eindickenden verlustfreien Datenentsafter. Das an-

schließende Verdünnen des Sirups war aber relativ umständlich und langsam. Die Folge: Man kann auf die Thompson-Daten nur in der Ausgangsstellung zugreifen. Ein Zugriff im Baum bringt die Suche zum Stillstand. Damit war der praktische Wert dieser Daten sehr stark eingeschränkt. Befindet man sich einmal in einem 4- oder 5-Steiner, dann sind die Programme meist mit Hilfe nackter Gewalt in der Lage, die Gewinnführung zu errechnen. Praktisch relevante Ausnahmen sind eigentlich nur »König und Dame gegen König und Turm« und »König und Turm gegen König und Springer«. Es passierte nicht nur einmal, daß ein mit den Thompson-CDs ausgerüstetes Programm frohgemut in ein Endspiel abwickelte und nach erfolgreichem Abtausch feststellte, daß es nun in 20 Zügen mattgesetzt wird. Manchmal weigerte es sich auch standhaft, in ein leicht gewonnenes Endspiel abzutauschen. Programme werden durch diese Datenbanken bestenfalls um ein Elo besser. Die Benutzer fanden aber Gefallen daran. John Nunn hat seine KTBKT-Datenbank-Erfahrungen sogar zu einem eigenen Buch verarbeitet.

Mollig ist sexy

Die Thompson-Datenbanken waren wegen des relativ komplizierten Entpackungs-Verfahrens bei den Programmierern nicht besonders beliebt. 1994 begann S. Edwards mit der Konstruktion seiner »Tablebases« genannten Endspiel-Datenbank. Er opferte die Komprimierung zugunsten einer idiotensicheren Zugriffsmethode. Edwards konstruierte alle 3-, 4- und die wichtigsten 5-Steiner. Nach Edwards Tablebase-Pensionierung wurde die Arbeit vom Microsoft-Angestellten Eugene Nalimov übernommen. Nalimov vervollständigte zunächst die 5-Steiner. Zur Zeit sind sogar einige »kleine« 6-Steiner wie KDD-KDD in Bearbeitung. KDD-KDD ist ein kleiner 6-Steiner, weil man zusätzlich zur Figuresymmetrie die erste weiße und schwarze Dame auf das Dreieck a1-h1-h8 einschränken kann.

Nalimov führte auch eine dynamische Komprimierungs- bzw. Entpackungsmethode ein. Die Nalimov-Tablebases werden dabei mit einem einfachen und schnellen, aber dennoch relativ effektiven Verfahren komprimiert. Beim Zugriff auf eine bestimmte Position wird immer ein ganzer Abschnitt der Datenbank entpackt und in den Hauptspeicher geladen. Bei einem weiteren Zugriff überprüft das Programm zunächst, ob die Stellung nicht bereits im entpackten Abschnitt enthalten ist. Ist dies der Fall, genügt ein schneller Tabellenzugriff im Arbeitsspeicher. Ist die Stellung im Endspiel-Cache noch nicht enthalten, wird wieder auf die Festplatte zugegriffen, der Abschnitt entpackt und in den Hauptspeicher geladen. Falls der für den Endspiel-Cache reservierte Hauptspeicher bereits voll ist, wird der bisher am wenigsten benützte Datenbank-Teil durch den neuen Abschnitt überschrieben.

Cache-Roulette

Mit Hilfe dieses Cache-Verfahrens kann man auch in der Suche auf die Endspieldatenbank zugreifen. Dem Programm wird damit die Möglichkeit genommen, sich an der harten Wirklichkeit vorbeizuschwindeln, bzw. es wird einen Abtausch in ein gewonnenes Endspiel sofort ausführen. Das Verfahren ist aber nicht ohne Risiko. Solange ein Programm in seiner Suche immer in dieselben Datenbank-Abschnitte abwickelt, funktioniert alles bestens. Nach ein paar Zugriffen auf die Festplatte ist das Cache mit den richtigen Positionen gefüllt, die Suche erhält ungebremst die perfekten Endspiel-Bewertungen. Nachdem ein Programm in seiner Suche Schlagzügen den Vorrang gibt, wickelt es aber speziell in tieferen Suchen in alle (un-)möglichen Endspiele ab. In diesem Fall ist es primär mit dem Nachladen von neuen Abschnitten beschäftigt. Im Informatiker-Jargon nennt man das Cache-Trashing. Die Suchgeschwindigkeit bricht im Extremfall auf ein paar hundert Stellungen pro Sekunde ein. Um diesen Effekt zu vermeiden, schränken viele Programmierer den Zugriff auf die ersten 7 bis 8 Halbzüge ein. Das ist ein sehr guter Kompromiß – allerdings erkennt das Programm damit keine tiefen Abtäusche und eröffnet damit so manchen Schummelpfad.

Klein, aber fein

Nach den gescheiterten König, Dame gegen König, Bauer auf c2-Versuchen stellte ich mir die Aufgabe, für Nimzo7.32 eine stark komprimierte Datenbank zu erstellen, die man permanent in den Hauptspeicher laden kann. Konkret sollten alle 4-Steiner nicht mehr als 10 MByte Speicher benötigen, der Zugriff auf eine Position sollte mindestens so schnell sein wie der Aufruf der entsprechenden Bewertungsfunktion. Herkömmliche Datenbanken speichern die Distanz zum Matt ab und benötigen pro Position ein Byte. Für den korrekten Abtausch in ein Endspiel genügt aber die Information Gewonnen, Remis, Verloren. Befindet sich das Programm einmal in einem gewonnenen Endspiel, kann man noch immer die Nalimov-Tablebases oder auch die Thompson-Daten abfragen. Durch die Reduktion auf diese drei wesentlichen Werte benötigt man pro Position nur mehr 2 Bits. Man kann also 4 Positionen in einem Byte abspeichern (es gehen sogar 5 Positionen pro Byte). Meistens ändert sich durch eine geringfügige Abwandlung der Stellung am Resultat nichts. Benachbarte Bytes haben daher mit hoher Wahrscheinlichkeit denselben Wert.

Speziell bei 256-Farben-Grafiken tritt derselbe Effekt auf. Benachbarte Pixel haben oft dieselbe Farbe. Um nicht das Rad neu zu erfinden, studierte ich daher die gebräuchlichsten Grafikformate. Am besten gefiel mir die verlustfreie Komprimierungsmethode von McPaint. Grafikformate komprimieren meist zeilen-

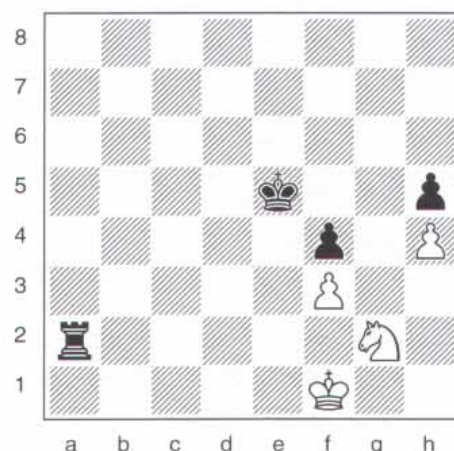
weise. Dies hat den Vorteil, daß sie zum Darstellen einer bestimmten Zeile nicht das gesamte Bild, sondern nur die Zeile selber entpacken müssen. Bei größeren Bildern kann man dadurch z.B. nur jede vierte Zeile entpacken und so dem Benutzer einen ersten schnellen Eindruck vom Bild geben. Außerdem pflanzen sich Fehler (etwa bei Übertragung via Internet) nicht in die nächste Zeile fort.

In Nimzo7.32 wird analog jedes Endspiel in 512 Byte lange »Zeilen« zerlegt. Aus der Stellungsinformation wird zunächst die »Zeilennummer« berechnet. Anschließend wird die Zeile bis zum Auffinden der jeweiligen Position entpackt und die Bewertung (gewonnen, remis oder verloren) gelesen. Mit dem McPaint-Komprimierungsverfahren kann man sehr schnell den nicht verwendeten Beginn der Zeile überspringen. Auf meinem 166 MHz-Entwicklungsrechner werden auf diese Art und Weise ca. eine Million Stellungen pro Sekunde gelesen. Nachdem sich das Programm durch den Zugriff auf die Datenbank die normale Bewertung erspart, wird es sogar etwas schneller. Zusätzlich zu den »normalen« 4-Steinern erzeugte ich noch die Pseudo-4-Steiner KBB-KB und KBB-KBB mit blockierten Bauernpaaren. Das paßte tatsächlich alles in die vorgesehenen 10 MByte. Bei der Weltmeisterschaft in Paderborn hatten wir einen Rechner mit 512 MByte Hauptspeicher zur Verfügung. Um diesen Speicher ordentlich zu füllen, produzierten wir noch zusätzlich die wichtigsten 5-Steiner im NCD (Nimzo-Compressed-Data)-Format.

Was bringt es?

Die Alexschen harten Nimzo-Nüsse wurden mit Hilfe der NCD-Datenbank sehr weich.

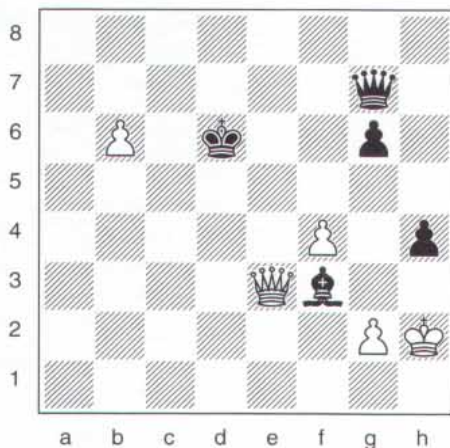
Geller, E. – Mikhalchishin, A.
Riga 1985



Schwarz am Zug

Diese Stellung löst Nimzo 7.32 in ein paar Mikrosekunden. Nach dem Abtausch **1...Txf2 2. Kxf2** befindet es sich bereits in seiner KBB-KBB mit blockierten Bauern Endspiel-Datenbank.

**Geller, E. – Fischer, R.
Havana 1965**



Weiß am Zug

Auf 1. **De5+** in dieser Stellung muß man schon wesentlich länger warten. Hier benötigt das Programm noch ca. 1 Sekunde bis zum Finden des besten Zuges.

Teststellungen und Turnierpartien sind aber zwei Paar Schuhe. Eine Endspiel-Datenbank hat auf das Ergebnis keinen Einfluß, wenn das Spiel im Mittelspiel bereits entschieden wurde. In einigen Endspielen wird der Gewinnweg durch die Datenbank zwar verkürzt, das Programm würde auf verschlungeneren Pfaden aber auch ohne Datenbank gewinnen. In diesem Fall wirkt sich die Datenbank ebenfalls nicht auf die Elowertung aus. Wieviele Partien letztendlich durch die Datenbank entschieden werden, läßt sich schwer sagen. Wir haben es nicht exakt gemessen, und ich kenne auch keine Ergebnisse von anderen Schach-Programmierern. Es wäre ein interessantes Experiment, wenn die Schweden auch einen Nimzo 7.32 ohne Datenbank in die Wertung aufnehmen würden.

Der perfekte Autist

Umfangreiche Erfahrungen mit Endspiel-Datenbanken hat nur Jonathan Schaeffer mit seinem Dameprogramm *Chinook* gesammelt. *Chinook* ist offizieller Mensch- und Maschine-Damechampion. Es war in der Endausbaustufe mit einer 8-Steiner-Endspieldatenbank ausgestattet. Die Zugriffsmethode war eine Mischung aus Nalimov-Cache und NCD-ähnlicher Komprimierung. Nachdem in Dame Schlagpflicht herrscht, gelangt man in diesem Spiel wesentlich schneller ins Endspiel. In vielen Partien rechnet *Chinook* daher direkt vom Eröffnungsbuch in die Endspieldatenbank. In gewisser Weise führte es in diesen Partien keinen einzigen »eigenen« Zug aus. Zur Überraschung des *Chinook*-Teams verbesserte sich der Punkteschnitt gegen menschliche Gegner durch die mächtige Datenbank nicht wesentlich. Das Programm wurde zwar beinahe unschlagbar, aber es gewann auch selber weniger Partien als ohne 8-Steiner. Von einem feurigen Angreifer mutierte es zum eher faden Remisschieber.

Programme sind Autisten. Sie spielen im Grunde immer gegen sich selbst. Ein Programm mit einer 8-Steiner-Datenbank nimmt daher an, daß auch der Gegner dasselbe perfekte Wissen hat. Für das Programm sind alle Endspielstellungen mit demselben Ausgang gleich gut. Für einen Autisten mit einem perfekten 8-Steiner gibt es keine schwierigen oder leichten Endspiele. Sie sind alle einfach. Das 8-Steiner-Chinook nahm daher dem Gegner oft die Chance, einen Fehler zu machen. Falls es keine Gewinnmöglichkeit sah, tauschte es einfach ins nächstbeste Remis-Endspiel ab. Obwohl Schaeffer über keinen derartigen Fall berichtet, kann ein mit einer mächtigen Datenbank ausgestattetes Programm dem Gegner sogar das Siegen erleichtern. Es sieht einen Gewinnweg für den Gegner. Um Matt in 30 zu vermeiden, macht es selber einen sehr schlechten Zug (z.B. opfert eine Figur). Der weniger perfekte Gegner wundert sich in diesem Fall über das für ihn überraschende Geschenk.

Diese Effekte treten aber primär im Spiel »Mensch gegen Maschine« auf. Beim Spiel zwischen Spitzenprogrammen herrscht meistens ein technologisches Gleichgewicht. In diesem Fall ist es daher durchaus sinnvoll anzunehmen, daß der Gegner ebenfalls perfektes Endspielwissen besitzt.

Spielmörder

Einfachere Spiele wie »Vier in einer Reihe« und »Mühle« wurden mit einer Kombination aus Endspieldatenbank und tiefer Berechnung bereits gelöst. Bei »Vier in einer Reihe« gewinnt immer der Anziehende, »Mühle« geht bei beiderseitigem perfekten Spiel Remis aus. Jonathan Schaeffer träumte zeitweise auch davon, Dame endgültig zu lösen. Ich persönlich bin kein Fan von diesen Ambitionen. Meiner Meinung nach ist ein gelöstes Spiel uninteressant und damit tot. Es taugt bestenfalls noch als Kinderspiel. Man sollte in seinem Forscherdrang sehr vorsichtig und behutsam mit Lebewesen aller Art umgehen. Schach steht auf Grund seiner Komplexität noch nicht auf dieser roten Liste der vom Aussterben bedrohten Spiele.

Allerdings hat bereits das Endspiel-Buch von John Nunn Diskussionen über die Ethik von Endspiel-Analysen mit Hilfe von Datenbanken ausgelöst. Ich habe zu diesem Thema auch die Meinung der beiden Modernisiererinnen eingeholt. Sie verstanden die Frage gar nicht. Das lag nicht an ihren fehlenden Schachkenntnissen. Es kam ihnen gar nicht in den Sinn, daß jemand jahrelang über ein Problem nachdenken wollte, anstatt per Mausklick die Antwort in Sekunden-schnelle präsentiert zu bekommen. Am drastischsten formulierte es meine Mutter: Wer jemals in seinem Leben einen vollen Wäschekorb mit der Hand gewaschen hat, weiß, daß der Fortschritt nicht aufgehalten werden sollte.