

# Eine Maschine als Schach-Großmeister

Vor 40 Jahren wurde erstmals die Konzeption eines Schachcomputers veröffentlicht. Seither haben solche Maschinen zunächst Anfänger, dann Meister und mittlerweile sogar schon Großmeister bezwungen. Wird Garri Kasparow der nächste sein?

Von Feng-hsiung Hsu, Thomas Anantharaman, Murray Campbell und Andreas Nowatzyk

Im Januar 1988 antwortete der Schachweltmeister Garri Kasparow auf die Frage, ob es einem speziell für das Schachspiel gebauten und programmierten Rechner noch vor dem Jahr 2000 gelingen könne, einen Großmeister zu bezwingen: „Ausgeschlossen – und sollte irgendein Großmeister doch Schwierigkeiten gegen solche Computer haben, bin ich gerne bereit, ihm ein paar Ratschläge zu geben.“

Zehn Monate nach dieser Bemerkung wurde der Großmeister Bent Larsen, ehemaliger Mitbewerber um den Weltmeistertitel, bei einem bedeutenden Turnier in Long Beach (Kalifornien) von einem Schachcomputer besiegt: von Deep Thought, den wir an der Carnegie-Mellon-Universität in Pittsburgh (Pennsylvania) entwickelt hatten. Der elektronische Rechner, eigens für diesen Zweck gebaut und mit spezieller Software versehen, gewann bei dem Turnier fünf weitere Partien, erreichte ein Remis und verlor ein Spiel. Durch dieses Ergebnis teilte er sich den ersten Platz mit Großmeister Anthony Miles. Da Maschinen nach den Turnierregeln keine Preisgelder erhalten dürfen, konnte Miles die Siegesprämie in Höhe von 10 000 Dollar alleine einstreichen. (Ein Jahr später wurde auch er in einem Schaukampf von Deep Thought bezwungen.)

Im Sommer 1990, als drei Mitarbeiter unserer Forschungsgruppe samt ihrer Maschine zur Firma IBM wechselten, hatte Deep Thought in zehn Partien gegen Großmeister die Hälfte, in 14 Spielen gegen internationale Meister 86 Prozent der maximalen Punktzahl erreicht (ein Sieg zählt einen Punkt, ein Remis bringt beiden Spielern je einen halben Punkt ein). Einige dieser Partien und

Dutzende von Spielen gegen weniger herausragende Gegner fanden unter der Schirmherrschaft des amerikanischen Schachverbandes statt, der daraufhin die Spielstärke der Maschine zu 2552 Punkten berechnete. Damit liegt sie in der unteren Hälfte des Großmeisterbereichs, während ein durchschnittlicher Turnierspieler es nur auf etwa 1500 Punkte bringt (Bild 2). Nachdem sie im August 1988 ihre gegenwärtige Analysegeschwindigkeit von 750 000 Stellungen pro Sekunde erreicht hatte, überschritt ihre Spielstärke 2600 Punkte.

Im Folgemodell dieses Computers, das irgendwann im Jahr 1992 spielfähig sein sollte, wird eine wesentlich leistungsfähigere Hardware die Analysegeschwindigkeit auf etwa eine Milliarde Stellungen pro Sekunde – mehr als das Tausendfache – hochtreiben. Das allein wird der Maschine eine größere Spielstärke verschaffen, als Kasparow oder irgendein Schachspieler der Geschichte je besessen hat.

## Die Herausforderung

Warum aber will man überhaupt einer Maschine beibringen, eine hölzerne Figur auf einem karierten Spielbrett in die Enge zu treiben? Erstens gilt Schach in der abendländischen Kultur seit jeher als das herausragende Verstandesspiel oder – mit den Worten Johann Wolfgang von Goethes – als der Probestein des Gehirns. Für einige Leute wäre eine erfolgreiche Schachmaschine der handgreifliche Beweis, daß der Akt des Denkens auf die Maschine übertragbar, für andere, daß Denken keine notwendige Voraussetzung des Schachspiels sei. Jede der

beiden Schlußfolgerungen würde sicherlich unser hergebrachtes Verständnis von Intelligenz grundsätzlich ändern.

Zum zweiten ist Computerschach eine technische Herausforderung ersten Ranges. Claude E. Shannon, der Begründer der Informationstheorie, formulierte sie schon vor 40 Jahren im Scientific American („A Chess-Playing Machine“, Februar 1950):

Die Untersuchung des Schachspiel-Problems zielt auf die Entwicklung von Techniken, die auch für praxisnähere Anwendungen genutzt werden können. Die Schachmaschine ist aus mehreren Gründen ein idealer Ausgangspunkt: Das Problem ist genau definiert, sowohl in bezug auf die zulässigen Operationen (die Züge) als auch auf das Endziel (Schachmatt). Es ist weder zu trivial noch zu schwierig für eine befriedigende Lösung. Außerdem kann man eine solche Maschine gegen einen menschlichen Gegner antreten lassen und so ihre Fähigkeiten im Denken dieser Art beurteilen.

Die vielleicht wichtigste praktische Konsequenz des Schachspiel-Programmierens ist der Nachweis, daß die maschinelle Analyse funktioniert. Der Stand der Perfektion, den diese und ähnliche Techniken erreicht haben, verspricht Gewinn in so unterschiedlichen Anwendungsbereichen wie dem Entwurf von Netzwerken oder der Modellierung chemischer Reaktionssysteme und sogar für linguistische Analysen abzuwerfen.

Die Idee einer schachspielenden Maschine geht bis in die sechziger Jahre des 18. Jahrhunderts zurück, als Baron Wolfgang von Kempelen seinen Schach-Automaten in Europa vorführte. Eine Puppe mit Schnurrbart und Turban,

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

„Türke“ genannt, bewegte die Figuren, allem Anschein nach von einem komplizierten Mechanismus in einem Kasten unter der Puppe gesteuert. Der Apparat spielte im allgemeinen gut; es gelang ihm, Napoleon Bonaparte in nur 19 Zügen mattzusetzen und damit einen Tobsuchtsanfall auszulösen.

Edgar Allan Poe erriet – wie auch weitere findige Köpfe – später das Geheimnis des Automaten: Ein zwergwüchsiger Schachmeister steuerte die Züge aus einem Versteck im Kasten. Nur war Poes Begründung, die auf Überschätzung der Technik beruhte, falsch: Die Tatsache, daß der Türke gelegentlich Partien verlor, sei mit der Perfektion einer echten Maschine unvereinbar.

#### Das Prinzip der Schachmaschine

Der britische Mathematiker, Informatiker und Kryptologe Alan M. Turing (1912 bis 1954) war einer der ersten, die

sich mit dem Problem eines schachspielenden Computers befaßten. Er fand es jedoch leichter, seinen einfachen Algorithmus, der Züge erzeugen und Positionen bewerten konnte, mit der Hand auszuführen als mit einem Rechenapparat. Der deutsche Computer-Pionier Konrad Zuse und andere unternahm ähnliche Anstrengungen, aber die grundlegende Arbeit stammt von Shannon. Er baute auf der Spieltheorie der (aus Ungarn beziehungsweise Deutschland stammenden) Amerikaner John von Neumann und Oskar Morgenstern auf, insbesondere auf dem sogenannten Minimax-Algorithmus, der einen optimalen Spielzug zu berechnen erlaubt.

Ein Schachprogramm erzeugt zunächst sämtliche Stellungen, die durch einen zulässigen Zug aus der gegenwärtigen Stellung entstehen können; auf jede dieser Stellungen wendet er jeden möglichen Zug des Gegners an, und so weiter. (In der Schachsprache heißt erst das Paar aus einem Zug von Weiß und einem

nachfolgenden Zug von Schwarz ein „Zug“; daher wird im folgenden die Einzelaktion eines Spielers als Halbzug bezeichnet.) Dann wird jede Endstellung durch eine Zahl bewertet und von dieser Bewertung ausgehend rückwärts gearbeitet, bis der optimale erste Halbzug bestimmt ist (Bild 3).

Jeder Halbzug in dem sich verästelnden Analysebaum umfaßt etwa 38mal so viele Stellungen (die typische Anzahl möglicher Halbzüge aus einer gegebenen Stellung heraus) wie die vorausgegangene Schicht oder etwa sechsmal so viele, falls Alpha-Beta-Pruning (eine Technik zum Beschneiden eines Suchbaums) verwendet wird. Daher ist die überwiegende Mehrzahl der Stellungen in den äußeren Knospen dieses Suchbaums repräsentiert, der von dort aus weiterwächst, bis entweder das Spiel zu Ende oder die selbstgesetzte Bedenkzeit abgelaufen ist.

Eine Bewertungsfunktion ordnet dann jeder Endposition einen Wert zu, etwa 1,

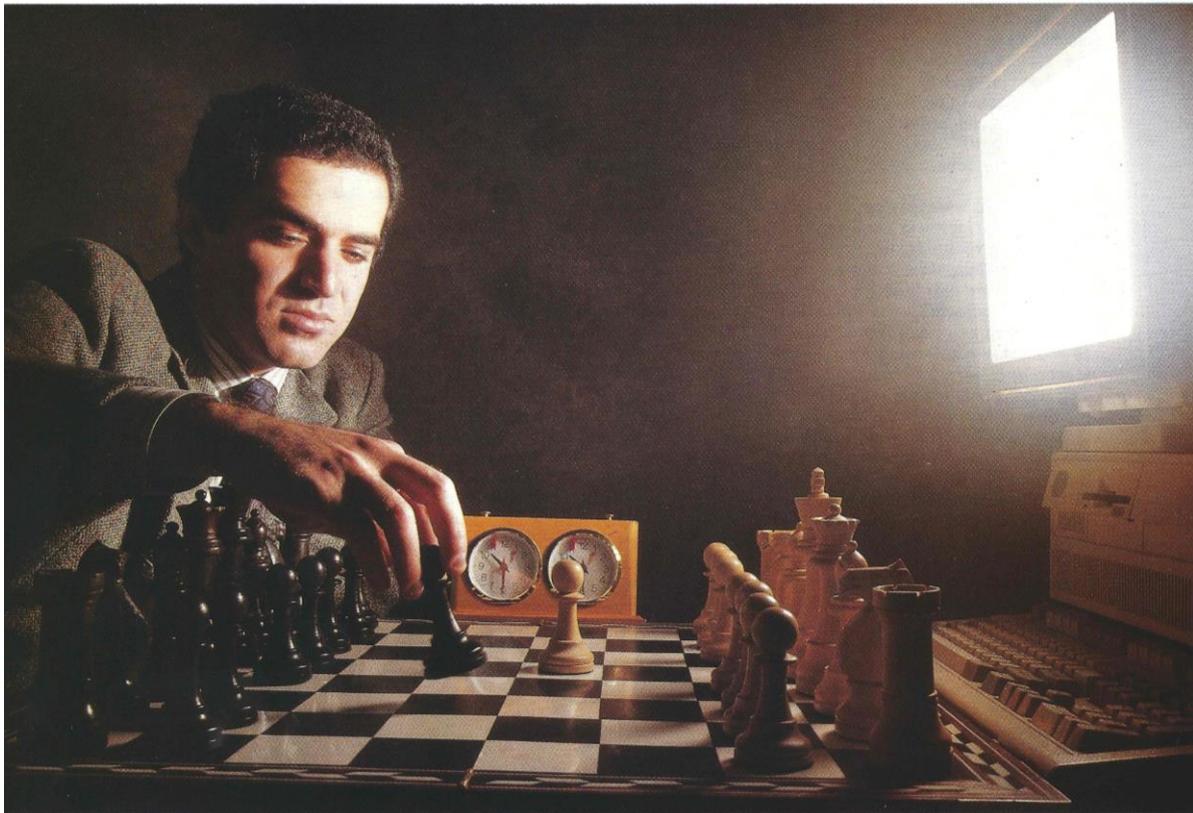


Bild 1: Schachweltmeister Garri Kasparow posiert mit einem Computer IBM PS/2, der für die Kommunikation mit dem elektronischen

Schach-Automaten Deep Thought verwendet wurde, vor Beginn seiner beiden Spiele gegen die Maschine im Oktober 1989. Kasparow gewann.

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

wenn der Gegner mattgesetzt ist, -1 im umgekehrten Fall und 0 bei einem Remis. Diese Bewertung kann erheblich verfeinert werden, indem etwa der Computer den Materialwert der Figuren bestimmt und den Wert einer Stellung nach gewissen Kriterien wie Figurplatzierung, Bauernstruktur, Kontrolle einer offenen Linie (einer vertikalen Linie von Feldern, die nicht durch Bauern blockiert ist) oder Kontrolle des Zentrums (der vier zentralen Felder des Schachbretts) berechnet.

Man kann die Spielstärke eines Computers steigern, indem man entweder die Geschwindigkeit erhöht, mit der er den Variantenbaum absucht, oder die Positionsbewertung verbessert. Ein Computer, der alle nur möglichen Spielvarianten erzeugt und dann die Endstellungen als verloren, gewonnen oder unentschieden klassifizieren könnte, würde nicht nur absolut fehlerfrei spielen; er könnte seinen Gegner bereits beim ersten Zug mit der Ansage verblüffen: „Weiß am Zug gewinnt in höchstens 137 Zügen“ oder aber aufgeben, weil er die Stellung als aussichtslos erkannt hätte. Derartig erschöpfende Analysen sind allerdings nur für so einfache Spiele wie Tic-Tac-Toe (vergleiche Computer-Kurzweil, Spektrum der Wissenschaft, Dezember 1989) leicht durchzuführen, aber für die etwa  $10^{120}$  verschiedenen Spielabläufe beim Schach nicht mehr praktikabel.

Andererseits wäre ein Computer ebenfalls unfehlbar, wenn er nur einen einzigen Halbzug vorausschaute – vorausgesetzt, die Stellungsbewertung wäre perfekt. Dies wäre eine Strategie nach dem Vorbild des Schachmeisters Richard Réti, der in den zwanziger Jahren auf dem Höhepunkt seiner Karriere stand. Er hatte scherzhaft behauptet, immer nur einen Zug weit vorausszusehen – den besten.

Solche hohen Ansprüche kamen jenen frühen Schachprogrammierern nicht in den Sinn, deren Programme erst 1958 überhaupt die Regeln des Schachspiels beherrschten. Weitere acht Jahre vergingen, bis das erste Programm, Mac-Hack-6 von Richard D. Greenblatt vom Massachusetts Institute of Technology in Cambridge (Massachusetts), die Stärke eines durchschnittlichen Turnierspielers erreichte.

Mit zunehmender Kopffzahl spaltete sich die Gruppe der Leute, die Schachprogramme entwickelten, in zwei ideologische Lager. Die einen – nennen wir sie die Emulierer – behaupteten, daß ein Schachprogramm nach den gleichen Prinzipien wie ein Mensch spielen sollte, etwa indem es seine Züge mit Hilfe expliziten Schlußfolgerns bestimmt. Die zweite Gruppe – die Ingenieure – vertrat die weniger restriktive Ansicht, daß eine

Denkweise, die gut für Menschen ist, nicht unbedingt gut für Computer sein muß.

Anfangs, als Computerschach noch mehr eine Frage der Theorie denn der Praxis war, hatten die Emulierer noch die Oberhand. Dies kehrte sich jedoch um, als man in den siebziger Jahren beobachtete, daß die Spielstärke der Programme ihrer Suchtiefe angenähert proportional war. Mit jedem zusätzlichen Halbzug legte der Computer etwa 200 Bewertungspunkte zu (Bild 2). Die Programmierer kämpften nun um den Zugang zu immer schnelleren Computern und erfanden immer neue Kniffe, um aus der vorhandenen Rechenleistung noch größere Suchtiefen herauszuquetschen.

### Die Kunst des Suchens

Richtig gesucht ist halb gewonnen. Ganz am Anfang der Schachprogrammierung erzeugten die Suchalgorithmen alle möglichen Spielabläufe mehr oder weniger blind. Sie behandelten gleiche Stellungen, die sich aus Zugfolgen in verschiedener Reihenfolge ergeben, als verschieden. Heute vermeiden sie diese Doppelarbeit, indem sie alle auftretenden Positionen in einer sogenannten Hash-Tabelle speichern. Diese sind noch wichtiger für den Alpha-Beta-Algorithmus, mit dessen Hilfe viele irrelevante Spielentwicklungen frühzeitig außer acht gelassen werden können.

Am schwierigsten ist es, im richtigen Moment mit der Erzeugung immer weiterer Suchbaumzweige aufzuhören. Man kann nicht alle Spielabläufe unbeschränkt weiterverfolgen, möchte jedoch andererseits die Analyse nicht in einer unstillen Stellung abbrechen, etwa wenn sich das Spiel mitten in einem Figurenabtausch befindet.

Nehmen wir an, daß der Computer alle Spielverläufe genau acht Halbzüge tief verfolgt und dabei eine Stellung findet, in der er einen Springer im Austausch für einen Bauern geschlagen hat. Selbst wenn der Gegner schon im nächsten Zug einen Springer zurückschlagen kann und dadurch um einen Bauern im Vorteil bleibt, wird der Computer unbeeindruckt auf den vermeintlichen Materialvorteil zusteuern.

Dieser sogenannte Horizont-Effekt kann Computer zu selbstmörderischen Zügen veranlassen, die selbst den schwächsten menschlichen Spielern nicht unterlaufen würden. Um solche Fehler zu vermeiden, fügt heute praktisch jedes Programm an die ursprüngliche Suche einen zusätzlichen Suchabschnitt an, die sogenannte Ruhe-Suche. Hier werden typischerweise nur solche Zugfolgen erzeugt, in denen Figuren ab-

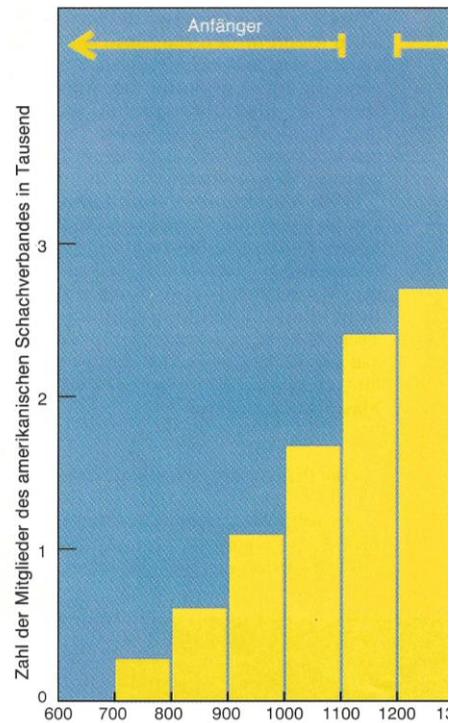


Bild 2: In der Spielstärke übertrumpfen die Computer bereits das Mittelfeld der Turnierspieler. Die gelben Balken zeigen an, wie sich die Spielstärken unter jenen 35 000 Mitglie-

getauscht werden, bis eine stabile oder ruhige, für eine Bewertung geeignete Stellung erreicht ist.

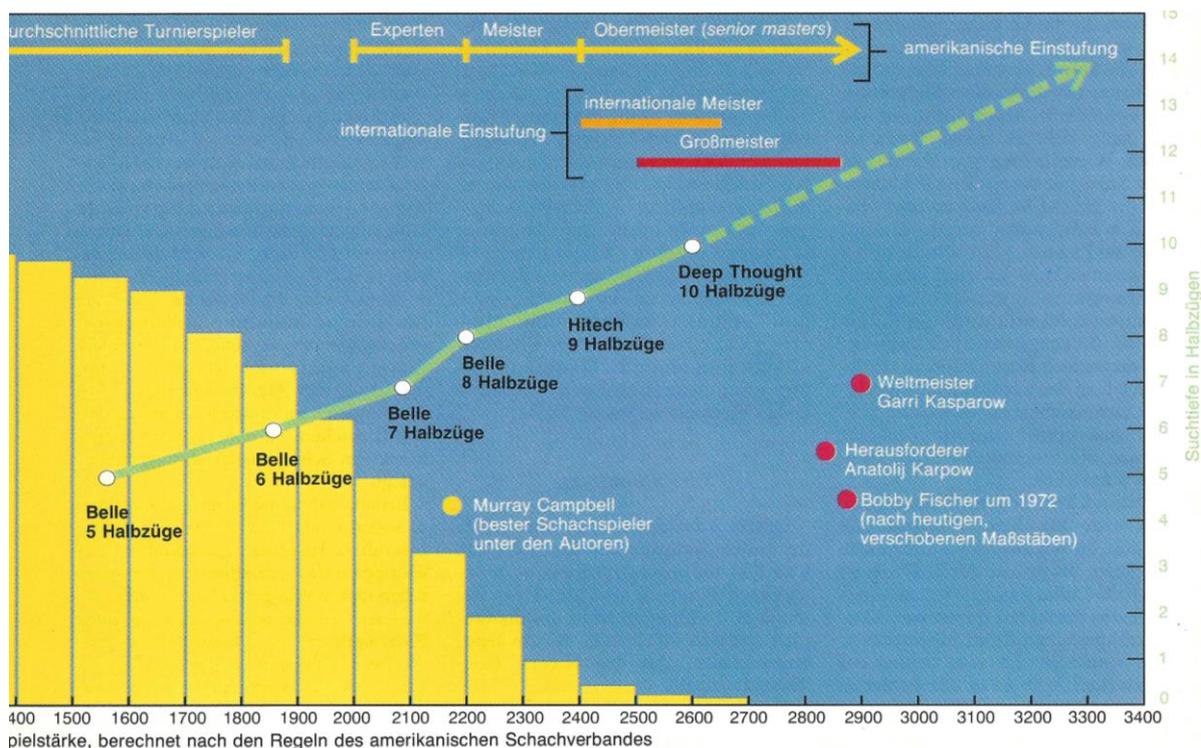
Die siebziger und frühen achtziger Jahre waren die Zeit der Kraftmeier-(brute force)-Programme und -Maschinen, die ihre Spielstärke im wesentlichen einer höchst effizienten Implementierung der Basis- und der Ruhe-Suche verdankten. Die Spitze hielten fast ununterbrochen das Programm Chess 4.0 und seine Nachfolgeversionen von der Northwestern University in Evanston (Illinois). Mit jeder Verbesserung der Hardware stieg auch die Spielstärke des Programms, bis es 1979 das Experteniveau von 2000 Punkten nach amerikanischer Wertung überschritt.

Ebenfalls schon aus den siebziger Jahren datieren einige eigens für das Schachspiel gebaute Computer. Der berühmteste, Belle aus den AT&T-Bell-Laboratorien in Murray Hill (New Jersey), überschritt 1983 die Grenze von 2200 Punkten, die nationale Meister von Experten scheidet.

Der Gipfel der Kraftmeierei waren 1986 Cray Blitz, ein Programm, das auf den Supercomputern der Firma Cray lief, und Hitech, eine Spezialmaschine,

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)



400 1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000 3100 3200 3300 3400  
 spielstärke, berechnet nach den Regeln des amerikanischen Schachverbandes

den des amerikanischen Schachverbandes (U.S. Chess Federation) verteilen, für die überhaupt solche berechnet wurden. Dem sind die Spielstärken von Maschinen mit un-

terschiedlichen Analysetiefen (grün) und einiger überragender Spieler (rote Punkte) gegenübergestellt. Mit jeder Vergrößerung der Suchtiefe um einen Halbzug steigt die Spiel-

stärke der Computer um etwa 200 Punkte. Deep Thought erreicht gegenwärtig etwa 2600 Punkte mit 10 Halbzügen. Sein Nachfolgemodell wird 14 oder 15 Halbzüge analysieren.

die mit Hilfe von 64 Chips – ein Chip für jedes Spielfeld – die Züge erzeugte. Hitech gewann 1985 die nordamerikanische Computerschach-Meisterschaft, unterlag aber Cray Blitz in der Computerschach-Weltmeisterschaft 1986 beim Stichentscheid in der letzten Runde (siehe Computer-Kurzweil, Spektrum der Wissenschaft, April 1986). Cray Blitz und Hitech untersuchten 100 000 beziehungsweise 120 000 Positionen pro Sekunde.

#### Die Anfänge von Deep Thought

Unsere Maschine hat eine recht ungewöhnliche Entstehungsgeschichte. Ersten waren ihre Entwickler einige Doktoranden ohne auswärtige Forschungsförderung oder direkte Anleitung von Wissenschaftlern der eigenen Hochschule (die Professoren an der Carnegie-Mellon-Universität, die sich mit Computerschach beschäftigten, hatten keine Verbindung zu unserem Team). Zweitens hatten die Mitglieder des Teams sehr unterschiedliche Vorbildungen, weswegen wir die Probleme auf sehr unorthodoxe Weise angehen.

Im Juni 1985 erkannte einer von uns (Hsu), daß man den Zuggenerator der Maschine Belle – den Teil der Hardware, der die möglichen Züge erzeugte – nach einigen Verbesserungen auf einem einzigen Chip in der Technik der Höchstintegration (*very large scale integration*, VLSI) unterbringen konnte. Die Forschungsabteilung des amerikanischen Verteidigungsministeriums (*Defense Advanced Research Programs Agency*, DARPA) machte damals den Universitäten diese Technik über Fördermittel zugänglich. Hsu entwarf auch den Chip mit der dichten Packung aller Bauteile einschließlich der 35 925 Transistoren; allerdings konnte MOSIS, die Lieferfirma für die von der DARPA finanzierten Chips, nur eine ziemlich geringe Auflösung von drei Mikrometern anbieten. Nachdem er sechs Monate mit dem Entwurf, dem Austesten und dem Anordnen der Komponenten zugebracht hatte, mußte er noch vier Monate warten, bis er das erste funktionsfähige Exemplar in Händen hielt. Er testete den Chip und fand, daß er bis zu zwei Millionen Züge pro Sekunde bearbeiten konnte – zehnmal so viele wie das 64-Chip-Modul von Hitech.

Zu diesem Zeitpunkt tat sich Hsu mit Thomas Anantharaman zusammen, der damals als Informatikstudent an einem Projekt der Universität für Spracherkennung mitwirkte. Anantharaman hatte ein kleines Schachprogramm geschrieben, das seine Züge mit Hilfe eines Softwarepakets erzeugte. Indem er dieses Paket durch Hsus Chip-Tester ersetzte, konnte er die Analysegeschwindigkeit des Programms auf 50 000 Stellungen pro Sekunde versechsfachen.

Nun packte Hsu und Anantharaman der Ehrgeiz; und spaßeshalber entschlossen sie sich, ihre Maschine für die nordamerikanische Schachcomputer-Meisterschaft 1986 vorzubereiten, die in nur sieben Wochen stattfinden sollte. Auch Murray S. Campbell und Andreas Nowatzky, zwei Informatikstudenten, ließen sich für das Projekt begeistern.

Ein besserer Bewertungsalgorithmus schien noch erforderlich – Campbell, der früher Turnierschach gespielt hatte, widmete sich diesem Problem. Die zweite und unter dem Zeitdruck noch schwierigere Aufgabe bestand darin, die nur zu Testzwecken aufgebaute Kombination aus Chip und Workstation zu einer vollständigen, einfachen Suchmaschine zu

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

erweitern, die das Leistungspotential des Chips besser ausschöpfen würde.

Um den Termin einzuhalten, entschloß sich Hsu zu der drastischen Maßnahme, zwei wesentliche Aspekte des Schachspiels außer acht zu lassen: Rochaden und die Wiederholung von Stellungen. (Jeder Spieler kann ein Remis fordern, wenn die gleiche Stellung dreimal wiederholt wurde, wobei jeweils derselbe Spieler am Zug war.) Zur Milderung der Folgen entwickelte er eine gemischte Suchstrategie: Die ersten Halbzüge wurden unter Berücksichtigung von Rochaden und Stellungswiederholungen auf dem sogenannten Host-Computer durchgerechnet, an den wiederum die eigentliche Schachmaschine angeschlossen war. Diese analysierte dann die späteren Halbzüge, die den größten Teil des Suchbaums ausmachen.

Wir hatten kein Geld und bauten daher unsere erste Maschine, ChipTest, mit Bauteilen, die wir aus anderen Projekten abzweigten. Insgesamt dürfte ihr Wert unter 1000 Dollar betragen haben, einen geschätzten Stückpreis für die von DARPA bereitgestellten Chips eingerechnet. Leider hatten wir bis zum Beginn der Meisterschaft noch nicht alle Fehler in der Hard- und Software ausgemerzt, und so mußte sich die Maschine mit einem Platz im Mittelfeld bescheiden – immerhin kein schlechtes Ergebnis für nur sieben Wochen Arbeit.

Wir lernten sehr viel aus diesem Debüt. Hsu beobachtete zum Beispiel, daß zwei andere Programme in eine Spielsituation geraten waren, in der jeder weitere Zug erzwungen wurde, aber keine Seite das Ergebnis vorhersehen konnte. Die Maschine, die sich schließlich in der besseren Position befand, verdankte ihren Vorteil also schierem Glück. Hsu schlug vor, diesen unerwünschten Effekt durch einen Algorithmus namens *singular extension* („Weiterverfolgung einzelner Zweige“) zu vermeiden: In den Fällen, in denen der Computer nur eine einzige gute Entgegnung finden kann, dehnt er die Suchtiefe über das normale Maß hinaus aus mit dem Ziel, kritischen Situationen verschärfte Aufmerksamkeit zu widmen (Bild 3).

Falls einer der Spieler kurz davor steht, beispielsweise einen eingekesselten Läufer zu schlagen, findet der Verteidiger typischerweise immer weniger gute Gegenzüge, je größer die Suchtiefe wird. Gegen Ende gibt es nur noch eine gute Entgegnung; gleichwohl ist schließlich der Läufer verloren. Genau auf solche Fälle ist der Algorithmus der *singular extension* spezialisiert. In einem Falle gelang es dem Computer auf diese Weise, einen Schachmeister mit der Ankündigung „Matt in 19 Zügen“ aus der Fassung zu bringen.

Anantharaman, der einzige Mensch, der das von ihm geschriebene Programm für den Host-Computer verstand, programmierte auch die *singular extension* alleine. Derweil stellte Hsu den Mikrocode fertig, jenen Satz von Instruktionen, der die Hardware auf dem untersten Niveau steuert. Mit einer Suchgeschwindigkeit von 400 000 bis 500 000 Stellungen pro Sekunde spielte ChipTest auf der nordamerikanischen Computerschach-Meisterschaft 1987 die anderen Maschinen in Grund und Boden, darunter auch den Vorjahresweltmeister Cray Blitz. Damit war die Herrschaft der Kraftmeier-Maschinen am Ende. Heute stecken in fast jedem Spitzenprogramm zumindest einige Elemente selektiver Suche.

### Perfektionierung

Unsere Arbeiten hatten gezeigt, daß in der Geschwindigkeit der Hardware von ChipTest und in der Organisation der Suche noch Reserven steckten. Hsus Betreuer H. T. Kung spendete das Startkapital von etwa 5000 Dollar für ein neues Unternehmen, das den Namen Deep Thought erhielt.

Die Hardware von Deep Thought besteht in der Grundversion aus 250 Chips einschließlich zweier Prozessoren; diese passen auf eine gewöhnliche Platine, die etwa anderthalbmal so groß ist wie eine Seite dieser Zeitschrift (Bild 4). Auch diese Maschine wird durch ein Programm – die sogenannte Host-Software – gesteuert, das auf einer Workstation abläuft. Die Prozessoren sind nur geringfügig schneller als die von ChipTest, liefern jedoch durch die verbesserte Steuerung des Suchalgorithmus eine 30 Prozent höhere Leistung.

Die Bewertungs-Hardware besteht aus vier Komponenten. Die erste (das einzige Erbstück von ChipTest) bewertet die Stellung eines Offiziers (einer Figur, die nicht ein Bauer ist) aufgrund der Felder, die er beherrscht, sowie seiner Beweglichkeit und einiger weiterer Kriterien. Die zweite Komponente berechnet entsprechend einen Wert für die Stellung der Bauern; hier geht ein, ob und inwieweit Bauern einander decken, das Zentrum beherrschen und den König schützen. Ein Freibauern-Bewerter ermittelt einen Wert für Bauern, die auf ihrem Weg zum anderen Ende des Spielbretts, wo sie sich in eine Dame umwandeln können, nicht mehr durch gegnerische Bauern behindert werden. Schließlich weist ein Linien-Bewertungsprogramm komplizierteren Konfigurationen aus Bauern und Türmen auf einer Linie einen Wert zu.

Die vier Komponenten der Bewertungs-Hardware liefern für jede vorge-

legte Stellung ungefähr 120 Zahlenwerte, die jeweils angeben, inwieweit eines von 120 wünschenswerten Zielen – eine Figur nicht zu verlieren, eine gegnerische Figur anzugreifen, eine eigene Figur zu decken und so weiter – bei der gegebenen Stellung erreicht ist. Jeder dieser Zahlenwerte ist sodann mit einem Parameter, einem sogenannten Gewicht, zu multiplizieren; die Produkte sind aufzuaddieren. Die Zahl, die sich daraus ergibt, ist der Wert einer Stellung. Auf diese Weise wägt die Maschine Vor- und Nachteile verschiedener Stellungen gegeneinander ab. Je höher das zu einem Ziel gehörige Gewicht ist, um so wichtiger schätzt der Algorithmus dieses Ziel im Vergleich zu anderen Zielen ein. In den Gewichten ist daher eine immense Menge an Schachspielerfahrung niedergelegt.

Bislang hatten die Programmierer diese sogenannten Gewichte nach Gutdünken durch Ausprobieren bestimmt. Wir überlegten uns Verfahren zur automatischen Optimierung und bauten sie – soweit wir wissen, als einzige – in unser Programm ein. Wir besorgten uns 900 Meisterpartien und definierten diejenigen Gewichte als optimal, bei deren Verwendung die Züge, welche die Maschine als die besten berechnete, am ehesten mit den tatsächlich gespielten übereinstimmten. Um diese Strategie zu implementieren, schrieben Campbell und Nowatzky den Software-Teil der Bewertungsfunktion vollständig neu. Wenn das Bewertungsprogramm auf das Optimieren der Gewichte eingestellt ist, liefert es statt eines numerischen Wertes ein Gleichungssystem, in dem die Gewichte die Unbekannten (genauer: die Koeffizienten eines Optimierungsproblems) sind.

Zu dessen Lösung verwendeten wir zwei verschiedene Verfahren. Beim ersten, das wir als *hill climbing* (Bergsteigen) bezeichnen, weist man den Gewichten zunächst irgendwelche Werte zu und führt dann für jede Stellung aus der Spiele-Datenbank eine fünf oder sechs Halbzüge tiefe Suche durch. Auf der Basis der gegebenen Parameter wählt die Maschine zu jeder der Stellungen einen gewissen Zug. Der Optimierungsalgorithmus verändert dann einen der Parameter geringfügig und berechnet die Zugescheidungen auf der neuen Basis. Falls die Anzahl der Übereinstimmungen zwischen den Computer- und den Großmeister-Entscheidungen dabei zunimmt, wird der Parameter in der gleichen Richtung nochmals verändert. Dieser Prozeß wird so lange wiederholt, bis alle Variablen optimale Werte angenommen haben. Der vollständige Durchlauf des Verfahrens hätte allerdings Jahre gedauert; daher verwendeten wir es nur für einige schwierige Fälle.

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

Das zweite Optimierungsverfahren, von Nowatzky vorgeschlagen und implementiert, arbeitet wesentlich schneller. Man unterstellt, daß es so etwas wie eine perfekte Bewertungsfunktion gibt, und versucht, diese durch die von der Maschine berechnete Funktion möglichst gut anzunähern, das heißt die Summe der Quadrate der Abweichungen zwischen beiden Funktionen zu minimieren. Angenäherte Werte für diese hypothetische Funktion gewinnt man für diesen Zweck mit ausreichender Qualität, indem man Ergebnisse einer extrem tiefen Suche heranzieht (die ihrerseits durchaus mit einer nicht optimierten vorläufigen Version durchgeführt werden kann) oder die Entscheidungen der Maschine mit denen erstklassiger Schachspieler vergleicht.

Dem liegt die naheliegende Überlegung zugrunde, daß ein Zug, den ein Großmeister in einer gegebenen Situation ausführt, besser ist als jeder andere, den er möglicherweise erwogen und dann verworfen hat. Daraus schließt man, daß die perfekte Bewertungsfunktion für die gewählte Stellung einen höheren Wert hat als für die verworfenen; diese Differenzen gehen in das Optimierungsverfahren ein. Nowatzkys Algorithmus benötigt nur ein paar Tage und optimiert nicht einen Parameter nach dem anderen, sondern alle gleichzeitig.

Unsere automatisch optimierte Bewertungsfunktion scheint nicht schlechter,

vielleicht sogar besser zu sein als die sozusagen von Hand justierten Funktionen der bekannten akademischen Schachprogramme wie Hitech und Cray Blitz. Es klafft aber immer noch eine Lücke zwischen unserer Bewertungsfunktion und denen der besten kommerziellen Schachcomputer, die meist das Ergebnis vieler Mannjahre Arbeit sind. Wir hoffen jedoch, durch Verbesserung des Optimierungsverfahrens bald aufschließen zu können.

### Erfolge

Es mag seltsam anmuten, daß unsere Maschine, obgleich relativ wenig Wissen über das Schachspiel in sie eingebaut ist, exzellente menschliche Spieler übertreffen kann. Man muß jedoch bedenken, daß die Maschine nicht menschliches Denken nachahmt – sie erreicht das gleiche Ziel mit anderen Mitteln. Deep Thought schaut sehr weit voraus, nimmt aber wenig wahr, merkt sich alles, aber lernt nichts, macht keine Patzer, spielt aber auch niemals besser als sonst. Dennoch produziert die Maschine manchmal Einsichten, die selbst den besten Großmeistern nicht in den Sinn kommen.

Vielleicht waren diese nicht menschengemäßen Erkenntnisse der Grund für die Entscheidung des Großmeisters Kevin Spraggett, unsere Maschine als

Assistent bei der Vorbereitung für das Viertelfinale des Schachweltmeister-Kandidatenturniers gegen Großmeister Artur Jusupow einzusetzen. Die Teilnahme der Maschine hatte zwar keinen erkennbaren Einfluß auf das Spiel, war jedoch immerhin ein bemerkenswerter Präzedenzfall.

Bei einem Schaukampf in New York im Oktober 1989 absolvierte eine mit sechs Prozessoren ausgestattete experimentelle Version von Deep Thought zwei Spiele gegen Kasparow (Bild 1). Obwohl diese neue Version mehr als zwei Millionen Stellungen pro Sekunde absuchen konnte, hatte Kasparow keine Mühe, sie zu besiegen. Das Ergebnis selbst kam eigentlich nicht unerwartet, aber die Spielweise von Deep Thought war ziemlich enttäuschend.

Im vergangenen Februar spielte Deep Thought dann eine Partie gegen Kasparows Herausforderer Karpow. Wegen der Fehler, die sich in der experimentellen Software für die Vier- und die Sechs-Prozessoren-Version gezeigt hatten, griffen wir wieder auf das alte Zwei-Prozessoren-Modell zurück. Gestärkt durch eine Reihe von Verbesserungen in seiner Bewertungsfunktion, zeigte Deep Thought in den ersten 50 Zügen eines seiner besten Spiele; durch einen Patzer vergab es dann aber ein sicheres Remis. Eine stabile Sechs-Prozessoren-Version hätte über ausreichende Geschwindigkeit

Um aus einer gegebenen Spielstellung den besten Zug zu bestimmen, erzeugt ein Schachprogramm zunächst ein Sortiment möglicher erster Züge, dann – ausgehend von jeder der entstandenen Stellungen – ein Sortiment möglicher Züge des Gegners, von diesen aus wieder ein Sortiment eigener Züge und so weiter: Es entsteht ein vielfach verzweigter Baum; jeder Astabschnitt zwischen zwei Verzweigungen entspricht einem (eigenen oder gegnerischen) Halbzug. Das Programm bewertet sodann die Stellungen an den Enden der Zweige: Je vorteilhafter die Stellung aus der Sicht des Programms ist, desto höher ist die dieser Stellung zugeordnete Bewertungszahl. Im sogenannten Minimax-Algorithmus schreibt das Programm, von den äußersten Zweigen zur Wurzel rückwärts arbeitend, an jeden Verzweigungspunkt das Maximum der Bewertungszahlen der von diesem Verzweigungspunkt ausgehenden Aststücke, wenn diese eigenen Zügen entsprechen, und das Minimum, wenn es sich um Züge des Gegners handelt. Im ersten Beispiel (oben) erhält *B* die Bewertung 3, *E* die Bewertung -1 durch Minimierung; im nächsten Schritt wird *A* durch Maximierung mit 3 bewertet. Durch diese Auslese hat das Programm festgestellt, daß *D* die beste erreichbare Stellung ist – unter der Voraussetzung, daß der Gegner optimal spielt; es wird daher den Zug *B* wählen.

Um mit diesem einfachen Ansatz die Analyse einen Halbzug weiter zu treiben, müßte das Programm 38mal soviele Züge erzeugen und bewerten (in einer Schachposition gibt es durchschnittlich 38 Zugmöglichkeiten) und daher bei gleicher Zeitvorgabe 38mal so schnell sein. Geschicktes Beschneiden des Zugbaums (*alpha-beta pruning*) läßt irrelevante Spielverläufe von vornherein außer acht, womit der obige Faktor von 38 auf 6 absinkt. Falls im oberen Beispiel die Bewertung bei *C* anfängt und von dort aus nach rechts geht, wird das Programm *B* den Wert 3 zuweisen und dann feststellen, daß der Wert von *E* (wegen *F*) höchstens noch 2 sein kann und daher *G* nicht mehr untersucht werden muß.

Eine andere Methode, die selektive Weiterverfolgung (*singular extension*), nutzt die Möglichkeiten des Computers dadurch besser aus, daß sie sich auf kritische Stellungen konzentriert. Im unteren Baum hängt der Wert von *B* entscheidend vom Wert von *C* ab, während der Wert von *D* von keinem seiner Nachfolger abhängt. Um die Zuverlässigkeit der Bewertung von *A* zu steigern, würde das Programm daher *C* einen Halbzug tiefer untersuchen als sonst. Diese Methode ermöglicht es Deep Thought, in vielen taktisch komplizierten Stellungen extrem weit vorzuschauen.

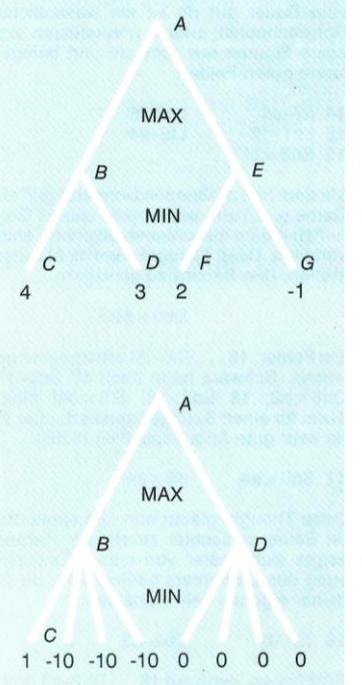


Bild 3: Der Minimax-Algorithmus zur Bestimmung des optimalen Zuges.

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

## Anatolij Karpow gegen Deep Thought

Harvard-Universität, Cambridge (Massachusetts), Februar 1990  
Kommentiert vom Internationalen Meister Michael Valvo

Weiß:  
Karpow

Schwarz:  
Deep Thought

1 e2-e4  
2 d2-d4  
3 Sb1-d2  
4 c2-c3  
5 e4-e5

c7-c6  
d7-d5  
g7-g6  
Lf8-g7

An dieser Stelle ist Deep Thoughts Eröffnungs-Lexikon, eine Sammlung abgespeicherter Spieleröffnungen, zu Ende.

... f7-f6!

Ein genialer Zug der Maschine. Schwarz greift sofort den Kopf der Bauernkette an und zwingt Weiß, seine Kontrolle über die weißen Felder zu schwächen.

6 f2-f4 Sg8-h6  
7 Sg1-f3 0-0  
8 Lf1-e2 f6xe5!

Deep Thought geht in die Offensive.

9 f4xe5 c6-c5!

Um nach 10 d4xc5 mit Sh6-g4! einen Positionsvorteil zu gewinnen.

10 Sd2-b3 c5xd4  
11 c3xd4 Sb8-c6  
12 0-0 Dd8-b6  
13 Kg1-h1 a7-a5

Weiß ist durch die zentrale Bauernphalanx immer noch leicht im Vorteil, doch sein Bauer auf d4 ist ein wesentlicher Schwachpunkt; zudem unterstützen sich seine Figuren nur schlecht und belegen keine guten Felder.

14 a2-a4 Lc8-f5  
15 Lc1-g5 Lf5-e4  
16 Sb3-c5!

Mit dem nun drohenden Doppelangriff auf Dame und Turm will Karpow seinen Gegner zu einem maschinen-typischen Fehler verleiten. Deep Thought kann nicht widerstehen, den Bauern zu schlagen:

... Db6xb2?

Ein Fehler; 16... Sh6-f5! wäre besser gewesen. Schwarz hätte nach 17 Sc5-d7, Db6xb2; 18 Sd7xf8, Sf5xd4! einen Turm für einen Springer geopfert, aber eine sehr gute Angriffsposition erzielt.

17 Sc5xe4 d5xe4

Deep Thought glaubt nun, um einen dritten Bauer schlechter zu stehen. Karpow zeigte sich später von dieser Einschätzung des Computers beeindruckt, da sie seiner eigenen sehr nahe kam.

18 Ta1-b1 Db2-a3

Erzwungen, denn auf 18... Db2-c3 droht 19 Tb1-b3, während Weiß auf 18... Db2-

a2 mit 19 Sf3-d2, gefolgt von 20 Le2-c4+ antworten würde.

19 Lg5-c1 Da3-c3  
20 Lc1-d2 Dc3-a3  
21 Ld2-c1 Da3-c3

Karpow wiederholt Züge, um Bedenkzeit zu gewinnen.

22 Tb1-b3 Dc3-a1  
23 Le2-c4+ Kg8-h8  
24 Lc1xh6! Da1xd1  
25 Lh6xg7+ Kh8xg7  
26 Tfxd1 e4xf3  
27 g2xf3

27 Tb3xb7 wäre besser gewesen. Aber wie hätte Karpow Deep Thoughts nächsten Zug voraussehen können?

... Ta8-7!!



Karpow

Die Zuschauer lachten über diesen vermeintlich „häßlichen“ Zug. Nach Karpows Ansicht war dies jedoch Deep Thoughts einzige Chance.

28 Lc4-d5 Tf8-d8  
29 Tb3-b5 Ta7-a6!

Der Computer verteidigt sehr geschickt. Nun droht er mit 30... Sc6-a7; 31 Ld5xb7, Sa7xb5; 32 Lb7xa6, Td8xd4 seinen Figurenrückstand aufzuholen.

30 Ld5-c4 Ta6-a7  
31 Lc4-d5 Ta7-a6  
32 Tb5-c5 Td8-d7  
33 Kh1-g2 Ta6-b6!  
34 Ld5xc6 b7xc6  
35 Kg2-f2!

Angesichts des leicht überlegenen Spiels von Schwarz ein riskanter Zug, aber Karpow will immer noch gewinnen.

... Td7-d5  
36 Tc5xd5 c6xd5

37 Td1-c1  
38 Kf2-e3

Tb6-b4  
Tb4xa4

Ein anderer und vielleicht einfacherer Weg zu einem Remis wäre 38... Tb4-b3+; 39 Ke3-e2; Tb3-b4 mit Stellungs-wiederholung, da Weiß seinen d-Bauern kaum aufgeben kann.

39 Tc1-c5 e7-e6  
40 Tc5-c7+ Kg7-g8  
41 Tc7-e7 Ta4-a3+  
42 Ke3-f4 Ta3-d3  
43 Te7xe6 Td3xd4+  
44 Kf4-g5 Kg8-f7!

Ein guter Verteidigungszug.

45 Te6-a6 a5-a4

Hier bestand die Chance zum Remis: 45... h7-h6+; 46 Kg5xh6, Td4-h4+; 47 Kh6-g5, Th4-h5+; 48 Kg5-f4, Th5-f5+; gefolgt von 49... Tf5xe5. Aber Deep Thought sieht sich im Vorteil.

46 f3-f4 h7-h6+  
47 Kg5-g4 Td4-c4?

47... g6-g5! hätte abermals Remis ergeben, aber die Maschine, die sich immer noch einen Vorteil errechnet, will den Bauern nicht für ihre Sicherheit opfern.

48 h2-h4 Tc4-d4  
49 Ta6-f6+ Kf7-g7  
50 Tf6-a6 Kg7-f7  
51 h4-h5 g6xh5?

Der entscheidende Fehler. Gegen 51... g6-g5 hätte Weiß nicht gewinnen können, wie Karpow später erläuterte.

52 Kg4-f5 Kf7-g7  
53 Ta6-a7+ Kg7-f8  
54 e5-e6 Td4-e4  
55 Ta7-d7 Te4-c4  
56 Td7xd5 h5-h4  
57 Td5-d3 Kf8-e7  
58 Td3-d7+ Ke7-f8  
59 Td7-h7 h6-h5  
60 Kf5-e5 h4-h3  
61 f4-f5 Kf8-g8  
62 Th7xh5 a4-a3  
63 Th5xh3 a3-a2

Ein menschlicher Spieler würde hier 63... Tc4-a4 spielen, um den Kampf zu verlängern, aber dann würde Weiß mit 64 Ke5-f6 alsbald das Ende herbeiführen.

64 Th3-a3 Tc4-c5+  
65 Ke5-f6 Schwarz gibt auf.

Deep Thought beurteilte seine Stellung als um mindestens sechs Bauernäquivalente schlechter als die von Weiß, so daß seine Erbauer an dieser Stelle das Spiel aufgaben. Die Maschine hatte zwar noch 20 Minuten Bedenkzeit, während Karpows Uhr nur noch weniger als eine Minute anzeigte; jedoch wäre es für ihn ein leichtes gewesen, den Sieg zu erzwingen.

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft - Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)

und damit Suchtiefe verfügt, um diesen groben Fehler zu vermeiden (siehe das kommentierte Protokoll Seite 100).

### Das nächste Modell

Geschwindigkeit ist die zentrale Zielvorgabe für die Maschine der nächsten Generation, die gegenwärtig im Thomas-J.-Watson-Forschungszentrum der IBM in Yorktown Heights (New York) entworfen wird. Sie soll tausendmal so schnell sein wie die gegenwärtige und damit mehr als eine Milliarde Stellen pro Sekunde untersuchen können – genug, um in den meisten Fällen 14 bis 15, bei erzwungenen Zugfolgen sogar 30 bis 60 Halbzüge vorausschauen. Falls die bisher beobachtete Beziehung zwischen Geschwindigkeit und Spielstärke weiterhin gilt, wird die neue Maschine 3400 Punkte erreichen, 800 Punkte über dem heutigen Leistungsstand von Deep Thought und 500 Punkte über der besten Bewertung von Kasparow.

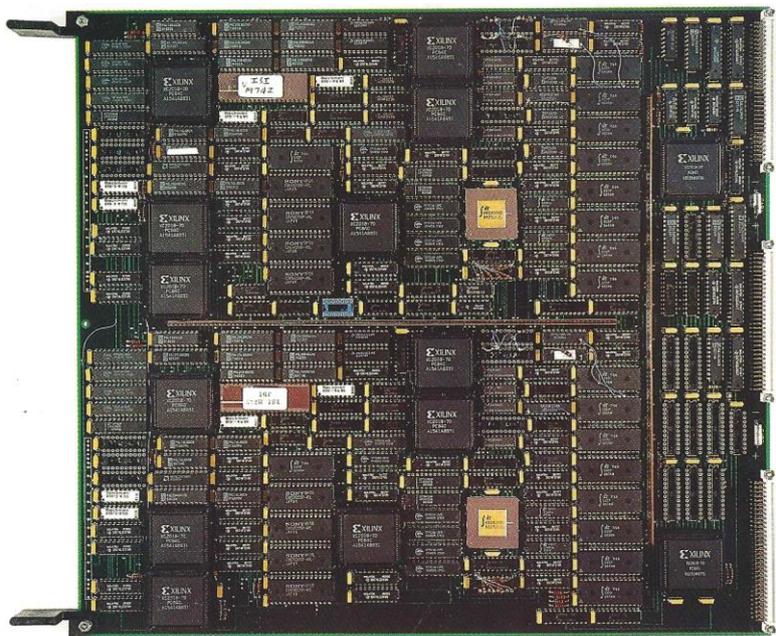
Zu diesem Zweck arbeitet Hsu an dem Entwurf eines schach-spezifischen Chips, der mit mindestens drei Millionen Zügen pro Sekunde dreimal so schnell sein soll wie die gegenwärtige Version von Deep Thought. Außerdem will er 1000 dieser Chips zu einem massiv parallelen Computersystem zusammenbauen, womit die Geschwindigkeit nochmals mindestens um den Faktor 300 steigen würde. Anantharaman und Campbell sind damit beschäftigt, verschiedene Aspekte der gegenwärtigen Version zu verbessern, die dann in die neue Maschine zu integrieren wären. Nowatzyk verfolgt mittlerweile andere Interessen.

Wir glauben, daß das neue System aufgrund seiner schieren Geschwindigkeit eine ernsthafte Herausforderung für den Schachweltmeister sein wird und sogar dank der langen Liste von geplanten Verbesserungen die Oberhand gewinnen kann – vielleicht schon 1992. Kasparow ist anderer Meinung, und wir müssen diese Ansicht respektieren. In einer privaten Mitteilung gestand er immerhin zu, daß eine Maschine, die eine Milliarde Positionen pro Sekunde untersucht, vielleicht einen durchschnittlichen Großmeister besiegen könne, „aber nicht Karpow und mich!“ Er vertritt die Meinung, daß die besten Spieler der Welt nach geeigneter Vorbereitung imstande sein müßten, die spezifischen Schwächen der Maschine zum Sieg zu nutzen, und daß die menschliche Kreativität und Phantasie im allgemeinen und seine eigene im besonderen mit Sicherheit über Silicium und Drähte triumphieren würden.

Auch wenn Kasparow verlieren sollte, hat er eigentlich keinen Anlaß zum – ihm ohnehin fremden – Selbstzweifel.

## Eine Maschine als Schach-Großmeister: Bericht über Deep Thought

(Quelle: Spektrum der Wissenschaft – Dezember 1990) (photo copyright © by [www.schaakcomputers.nl/](http://www.schaakcomputers.nl/)) (600 dpi)



**Bild 4:** Die Hardware von Deep Thought paßt auf eine Platine von der anderthalbfachen Größe dieser Seite. Jeder der beiden Prozessoren kann 500 000 Stellen pro Sekunde ab-

suchen. Im ab Ende 1992 einsatzbereiten Nachfolgemodell wird diese Rechenleistung auf einem einzigen Chip konzentriert sein; 1000 dieser Chips werden parallel arbeiten.

Schließlich würde dann die Genialität eines überragend begabten einzelnen Menschen der Arbeit von Generationen von Mathematikern, Informatikern und Ingenieuren gegenüberstehen.

Es wird nicht darum gehen, ob Maschinen denken können, sondern ob am Ende selbst die größten Erfolge der fähigsten Einzelnen gegenüber dem kollektiven Bemühen Vieler verblassen müssen.

**Feng-hsiung Hsu, Thomas Anantharaman, Murray Campbell und Andreas Nowatzyk** konstruierten Deep Thought, den besten Schachcomputer der Welt, während sie an ihren Promotionen in unterschiedlichen Gebieten der Informatik an der Carnegie-Mellon-Universität in Pittsburgh (Pennsylvania) arbeiteten. Hsu, Anantharaman und Campbell arbeiten inzwischen am Thomas-J.-Watson-Forschungszentrum der IBM in Yorktown Heights (Bundesstaat New York); Nowatzyk ist nun für die Computerfirma Sun Microsystems tätig. Hsu, der Initiator und Systemarchitekt des Projekts, hat einen Hochschulabschluß in Elektrotechnik von der Nationalen Universität von Taiwan. Anantharaman, Elektrotechnik-Absolvent der Hindu-Universität in Benares (Indien), schrieb den größten Teil der Host-Software. Campbell betreute das Schacheröffnungen-Lexikon und schrieb die ersten Stufen der Software für die Bewertungsfunktion; er hatte seinen ersten Hochschulabschluß in Informatik an der Universität von Alberta in Edmonton

(Kanada) gemacht. Nowatzyk, der in Hamburg sein Diplom in Physik und Informatik erwarb, entwarf und implementierte die automatische Parameteranpassung der Bewertungsfunktion.

### Literaturhinweise

Belle. Von J.H. Condon und Ken Thompson in: Chess Skill in Man and Machine. Zweite Auflage. Herausgegeben von P.W. Frey. Springer-Verlag, 1984.

Chess 4.5 – The Northwestern University Chess Program. Von David J. Slate und Lawrence R. Atkin in: Chess Skill in Man and Machine. Zweite Auflage. Herausgegeben von P.W. Frey. Springer-Verlag, 1984.

Large-Scale Parallelization of Alpha-Beta Search: An Algorithmic and Architectural Study with Computer Chess. Dissertation von Feng-hsiung Hsu. Carnegie-Mellon University, Computer Science Department, Preprint Nummer CMU-CS-90-108, Februar 1990.