KERNFORSCHUNGSZENTRUM

KARLSRUHE

Februar 1972 KFK 1506

Institut für Reaktorentwicklung

Über die Struktur sensori-motorischer Automaten

D. Smidt

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

Februar 1972 KFK 1506

Institut für Reaktorentwicklung

Über die Struktur sensori-motorischer Automaten

D. Smidt

Gesellschaft für Kernforschung mbH., Karlsruhe

Zusammenfassung

Die Grundzüge eines Rechnersystems werden konzipiert, das aus der Umwelt durch Sensoren eine komplexe Musterinformation empfangen und seinerseits den Zustand der Umwelt über Manipulatoren verändern kann. Es werden Verfahren entwickelt, die unter Benutzung der Wechselwirkung Rechner - Umwelt dem System über Lernprozesse und ohne gezielte äussere Eingriffe erlauben

- relevante Muster herauszufiltern,
- Ähnlichkeiten von Mustern in bezug auf verschiedene Transformationen zu erkennen und Zuordnungen nach verschiedenen Merkmalen auszuführen,
- die Zahl der zur Klassifizierung benötigten Musterkomponenten durch Generalisierung und Differenzierung zu optimieren,
- heuristische Prozesse durchzuführen,
- aus einfacheren schrittweise mehr und mehr zusammengesetze kognitive Prozesse und Aktionen aufzubauen,
- rückwirkend auch in späteren Stufen die primäre Kodierung der sensorischen Information zu verbessern.

Solche selbstorganisierenden, ihren Vorrat an Algorithmen in Wechselwirkung mit der Umwelt selbst definierenden Automaten werden in Anlehnung an das menschliche Verhalten auf der sensorimotorischen Intelligenzstufe "sensorimotorische Automaten" genannt.

Abstract

The basic concept of a computer system has been developed, which receives from the environment through sensors a complex pattern information and is able to change the environment by manipulators. Procedures have been outlined, which enable the system by use of the interaction computer - environment through learning processes without guidance

- to filter out relevant patterns,
- to recognize similarities of patterns with respect to different transformations and to arrange them with respect to different features,
- to optimize the number of pattern components necessary for classification by generalization and differentiation,
- to execute heuristic processes,
- to construct stepwise more composite cognitive processes and actions from simpler ones,
- to improve retroactively also from later steps the primary encoding of sensoric information.

Such automata, selforganizing and self-defining their store of algorithms, will be called "sensorimotoric automata" in relation to human behaviour on the sensorimotoric stage of intelligence.

Über die Struktur sensori-motorischer Automaten

Inhalt:

1. Einleitung

Technische Prozesse lassen sich im Prinzip dann automatisieren, wenn es ein auf einem Rechner handzuhabendes mathematisches Modell gibt, das sie hinreichend genau beschreibt und in dem die Prozeßvariablen und das Prozeßziel quantitativ definiert sind. Die Prozeßvariablen müssen meßbaren Größen des Prozesses zugeordnet sein. Im allgemeinen wird die Erstellung und Handhabung des mathematischen Modells um so eher möglich sein, je geringer die Zahl der beteiligten (und meßbaren) Variablen ist und je weniger das Prozeßziel Änderungen unterworfen werden kann und darf, also je weniger flexibel der Prozeß sein muß.

Es gibt nun viele Prozesse, bei denen die bis heute entwickelten Automatisierungsmethoden versagen, weil die von den Meßfühlern gelieferte Information zu umfangreich und vielgestaltig ist, als daß sie sich in einem realisierbaren mathematischen Modell verwenden ließe. Das gilt insbesondere dann, wenn die Information nicht einzelnen Variablen, also Drücken, Temperaturen oder Helligkeitswerten, sondern Variablenmustern, vor allem Mustern im optischen oder akustischen Bereich, zuzuordnen ist. Die Information wird besonders mannigfaltig, wenn der Prozeß in einer "natürlichen" Umwelt stattfindet, in der die relevanten Muster in einer nahezu unendlichen Vielzahl von irrelevanten Mustern enthalten sind. Gerade hier aber wird für die Prozesse auch noch häufig eine große Flexibilität gefordert, die die Modellerstellung weiter erschwert.

Beispiele hierfür sind das Aufführen einer Ziegelmauer, die Handhabung von Gepäckstücken an Flughäfen, die Führung eines Krans u. ä. Besonders interessant sind aber auch die in den "Heißen Zellen" der Kerntechnik durchzuführenden Operationen. Bei der Untersuchung oder Verarbeitung etwa eines bestrahlten Brennelementes müssen viele an sich jeweils "einfache" Verfahren wie Sägen, Schleifen, Ätzen,

---

zum Druck eingereicht am 4. 2. 1972

Auflösen, Fotografieren, Wiegen, Drehen, Fräsen, Bohren, Herstellen von Röntgen-Beugungsbildern usw. in wechselnder Reihenfolge angewandt werden. Hierfür steht neben den Spezialmaschinen als sehr universelles Werkzeug der Manipulator zur Verfügung. Da Heiße Zellen einen erheblichen Investitionswert darstellen, ist ein effizierter und in diesem Sinne automatischer Betrieb wünschenswert, scheitert aber zur Zeit an den technischen Möglichkeiten.

Wie muß nun eine solche Maschine, ein Prozeßrechner der nächsten Generation beschaffen sein, der Operationen von der beschriebenen Art durchzuführen gestattet? Offensichtlich muß sie die folgenden drei Fähigkeiten besitzen:

1. Sie muß Information, die als Muster auftritt, erkennen können. Über die heute schon entwickelten Mustererkenner hinaus muß sie aber auch die relevanten Muster aus vielen irrelevanten (oder im Augenblick jedenfalls irrelevanten) Mustern herausfiltern. Da bei der großen Fülle der möglichen Muster eine absolut strenge Zuordnung zwischen Mustern und Prozeßdaten vielleicht nicht durchführbar ist, sollte die Mustererkennung im Einzelfall eine größere Varianzbreite zulassen; d. h. die Klassifizierung aufgrund von Ähnlichkeiten oder die Generalisierung des Erkennungsprozesses sollte möglich sein.

2. Wegen der vielfältigen und komplexen Zusammenhänge zwischen den Prozeßvariablen sollte der Rechner selbst in der Lage sein, ein geeignetes Modell für den jeweils in Rede stehenden Prozeß herzuleiten. Das kann über ein lernfähiges System erreicht werden, das auch ohne Instruktor Informationen so speichert, daß sie für den nächsten spezifischen Bedarfsfall zur Verfügung stehen. Dabei bedeutet die Lernfähigkeit, daß nicht durch einen Instruktor eine punktweise Zuordnung zwischen einzelnen Informationen, Speicherplätzen und Bedarfsfällen vorgenommen wird, sondern daß das

System selbst Informationsstrukturen aufbaut, die die Zuordnung und Nutzung ermöglichen.

3. Der Rechner muß in der Lage sein, selbständig Ziele zu verfolgen und dazu seine gelernte Information zu benutzen. Zur Realisierung dieser Ziele müssen kontrollierte Aktionen vorgenommen werden, die Veränderungen der Umwelt bewirken.

In den folgenden Abschnitten soll die Konzeption eines EDV-Systems entwickelt werden, das Realisierungsmöglichkeiten für eine Prozeßdurchführung in dem beschriebenen allgemeinen Sinne bietet. Es würde in höherem Maße als konventionelle Rechner Fähigkeiten besitzen, die bisher nur dem Menschen oder einigen höher entwickelten Tieren zur Verfügung stehen. Andererseits wäre es aber auch noch weit davon entfernt, im eigentlichen Sinne "Intelligenz" zu besitzen. Eine Wechselwirkung zwischen System und Umwelt, bei der komplexe Informationsmuster wahrgenommen werden, die die Verfolgung konkreter Ziele durch unmittelbare motorische Aktionen auslösen und steuern, sei als sensori-motorisch bezeichnet, dem System sei entsprechend eine sensorimotorische Intelligenz zugeschrieben. Piaget [1, 2] bezeichnet die ersten zwei Lebensjahre des Menschen als die sensori-motorische Phase, in der im wesentlichen das Wahrnehmen und das konkrete Agieren mit Objekten eingeübt wird. Die Intelligenz der hier zu untersuchenden Automaten dürfte in der Tat am ehesten dieser Stufe zu vergleichen sein.

Damit werden die sensorimotorischen Automaten aber auch noch in ganz anderer, mehr fundamentaler Weise bedeutsam. Der Mensch entwickelt in den ersten Lebensjahren durch Lernprozesse die Grundlagen für sein späteres intelligentes Verhalten. Piaget und mit ihm fast alle Lernpsychologen vertreten die Hypothese, daß die komplexen Strukturen der Intelligenz aus einfacheren Schemata zusammengesetzt sind; diese

wiederum aus einfacheren und so herunter bis zu den zusammengesetzten und schließlich den Grundschemata der Sensorimotorik. Mit ihr übt der Mensch das Alphabet ein, aus dem die Sprache der intelligenten Prozesse aufgebaut ist. Wenn es gelingt, die Verhaltensformen der Sensorimotorik zu modellieren und damit zu experimentieren, wird sich eine größere Aussicht ergeben, auch die zusammengesetzten Abläufe zu synthetisieren und damit das Gebiet der künstlichen Ingelligenz im eigentlichen Sinne erfolgreich anzugeben.

Die Literatur hierzu zeigt, daß es bisher für diesen synthetischen Weg nur wenige Ansätze gibt. Auf allen Teilbereichen der "artificial intelligence", vor allem der Mustererkennung, der Sprachenübersetzung und der heuristischen Prozesse überwiegen die Versuche, für partielle Bereiche möglichst fortgeschrittene, "intelligenznahe" Strukturen zu entwickeln statt für den Gesamtbereich vorintelligende Bausteine einfacherer Art, eben das Alphabet an den Anfang zu stellen. Während der Mensch vermutlich nur mit einer einfachen Grundausrüstung von Algorithmen zur Informationsverarbeitung auf die Welt kommt und in der Wechselwirkung mit der Umwelt in fast zwei Jahrzehnten lernt, von einfachen konkreten zu komplexen abstrakten Problemen überzugehen, wird in den bisherigen Arbeiten zur künstlichen Intelligenz die Entwicklung von Algorithmen zur Lösung von Teilproblemen in den Vordergrund gestellt. Obwohl sich daraus viele sehr interessante Einsichten ergeben, mußte die eigentliche Frage unbeantwortet bleiben.

Das wird bei den heuristischen Verfahren besonders deutlich. Das verbreitetste Anwendungsbeispiel sind die zahlreichen Schach- u. Damealgorithmen (z.B. [3,4]). Sie sind aufgebaut auf eine vom Menschen durchgeführte Analyse des Spiels, eine daraus hergeleitete Strategie und ihre Übersetzung in die Sprache des Rechners. Entsprechende auf menschlicher Analyse aufgebaute Strategien lassen sich für andere Spiele ersinnen und programmieren bis hin zu Glücksspielen wie Poker. Solch

ein Verfahren bedingt eine geringe Flexibilität: Ein Programm, das mehr als ein Spiel spielen kann, würde einen erheblichen Aufwand bedeuten; und eines, das sich auf a priori nicht vorhersehbare Änderungen der Spielregeln einstellen kann, liegt außerhalb der Möglichkeiten. Ernst und Newell [5] haben mit ihrem General Problem Solver eine Verallgemeinerung der heuristischen Techniken erreicht und können damit die analytische Integration von Funktionen ebenso ausführen wie die Aufgabe von den Missionaren und den Menschenfressern, die in einem kleinen Boot über den Fluß gebracht werden müssen. Aber auch diese Programme bedürfen einer Analyse und Aufbereitung der Aufgabe und der Definition bestimmter heuristischer Kriterien durch den Menschen, es fehlen schlechterdings alle peripheren Organe zur direkten Wahrnehmung des Problems. Die Generalisierung schränkt zudem die Verwendbarkeit für Aufgaben wie das Schachspiel gegenüber den Spezialprogrammen stark ein.

Der Mensch dagegen erkennt und löst derartige Probleme zwar auch häufig durch ähnliche heuristische Such-Strategien, wie sie in solchen Programmen verwendet werden, doch setzt er dazu den gesamten Fundus seiner bis dahin erworbenen Informations- und Prozeßstrukturen ein und erfaßt die Aufgabe dadurch viel schneller und erzielt meist auch überlegene Lösungen. Darum sollten auch für die künstliche Nachbildung zunächst eher die grundlegenden Prozesse als die Strategien für spezielle Probleme untersucht werden.

Die künstliche Spracherkennung und -übersetzung wird ebenfalls über Algorithmen angegangen, die aus einer durch den Menschen vollzogenen Analyse entwickelt werden. Neben der Schwierigkeit, die sich hierfür durch den komplizierten Aufbau natürlicher Sprachen ergibt, spielt die Unmöglichkeit, den Kontext für die Entschlüsselung und Übersetzung aufgrund der Bedeutung der Werte auszunutzen, die wesentliche Rolle. Für die Wertbedeutung muß aber wieder auf alle früher gelernten Strukturen bis herunter auf ihre sensorimotorischen Elemente zurückgegriffen werden.

So ergibt sich also die Möglichkeit, neben der unmittelbaren Aufgabe, eine besondere Gruppe von technischen Prozessen zu automatisieren, das Problem der künstlichen Intelligenz von der Basis her neu aufzugreifen. Die hier zu entwickelnde Konzeption hat einige Gemeinsamkeit mit einem Entwurf von J. Becker [6], der jedoch die ersten Stufen der Verarbeitung der aus den Sensoren kommenden Information nicht behandelt. Gerade das wird hier ein wesentlicher Punkt sein, der die Struktur des Systems wesentlich bedingt. Auch in anderen Strukturbestandteilen und in den zu verfolgenden Strategien ergeben sich entscheidende Abweichungen.

Ein System, das die oben gestellten Prozeßaufgaben erfüllen kann, muß im Prinzip aus den folgenden Komponenten bestehen:

- Ein zur Mustererkennung geeigneter Sensor, z.B. eine Fernsehkamera,
- ein Kodierer, der die aufgenommenen elektrischen Signale in zur Weiterverarbeitung geeignete Datenstrukturen überführt,
- Speicher zur zweckmäßigen Aufzeichnung der brauchbaren Verknüpfungen zwischen Mustern und Aktionen,
- ein Rechenwerk, das die Organisation der gespeicherten Information, das Lesen der Speicherinhalte und die Ausführung der Aktionsbefehle steuert,
- einen möglichst vielseitigen Aktuator, z.B. einen in der Kerntechnik gebräuchlichen Manipulator, der dem System die Ausführung von Veränderungen in der Umwelt erlaubt.

  Dazu gehören Programme für die Informationsverarbeitung, Datenorganisation und das gesamte Lern- und Aktionsverhalten.
  Die erforderliche Hardware ist im Prinzip Stand der Technik. Auch die Speicher können von konventionellem Aufbau sein, wenn auch die Verwendung der heute noch in der Entwicklung befindlichen assoziativen

Speicher wesentliche Vereinfachungen im Speicherplatzbedarf und in der Systemorganisation erlauben werden.

Das System und die darin ablaufenden Prozesse sollen in der Folge im einzelnen behandelt werden.

2. Die Kodierung der Musterinformation aus der Umwelt

Aus den zahlreichen Arbeiten zur Mustererkennung (s. z.B. der in [7] und [8] gegebene Überblick) ist bekannt, daß die Musterinformation als Vektor (e_j) aus n Binärkomponenten e_j dargestellt werden kann. Im einfachsten Fall wird das in Rede stehende Gesichtsfeld in ein Raster aufgeteilt und jedem Rasterpunkt entspricht eine Komponente e_j; $e_j = 1$ heißt z.B., auf den Rasterpunkt fällt Licht, $e_j = 0$, der Rasterpunkt ist dunkel.

Aus Untersuchungen an Tieren ist bekannt (z.B. [9]), daß unmittelbar im Anschluß an das Sensorenraster, also etwa die Netzhaut des Auges, bestimmte abgeleitete Funktionen aus dem ursprünglichen Hell-Dunkel oder Ein-Aus-Muster gebildet werden.
So werden auf bestimmten Nervenfasern nur Signale detektiert, die etwa zeitlichen Intensitätsänderungen, räumlichen Intensitätsunterschieden bzw. Konturen im Gesichtsfeld, oder ersten und möglicherweise höheren räumlichen Ableitungen von Konturen in bestimmten Bereichen des Gesichtsfeldes entsprechen. Bei einzelnen Tieren sind sogar noch sehr viel weiter spezialisierte Detektoren gefunden worden, wie etwa der "Fliegendetektor" beim Frosch ([10], Lettvin u.a. "What the Frog's Eye Tells the Frog's Brain"), der nur auf dunkle konvexe Muster einer bestimmten Größe anspricht. Diese weitergehende Verarbeitung der Information vor ihrem Eintreten in die Nervenleitungen zum Gehirn ist offensichtlich bedeutsam. Im Interesse einer größeren Flexibilität muß allerdings dieser Prozeß nicht so weit getrieben und spezialisiert

werden, wie in dem zitierten Froschbeispiel. Nach Abschnitt 4.6 wird auch hier eine gewisse Lernfähigkeit vorzusehen sein.

Für die Kodierung optischer bzw. akustischer Musterinformation liegen in Anlehnung an biologische Vorbilder zunächst die folgenden Spezifikationen nahe:

a) Optisches Feld

a1) Grobes Raster mit Zuordnung jedes Rasterbereiches zu einer Komponente e_j. Die Helligkeit in jedem Rasterbereich wird in einer begrenzten Anzahl von Stufen angegeben, wozu weitere e_j benötigt werden.

a2) Feines Raster in einer kleinen Zone des Gesichtsfeldes (gelber Fleck), sonst wie a1, ergibt weitere e_j.

a3) Farbdetektor je Farbe ein e_j.

a4) Bewegungsdetektor, Bildung der zeitlichen Ableitung zwischen Zonen von Rasterpunkten; Besetzung weiterer e_j.

a5) Räumliche Helligkeitsunterschiede zwischen benachbarten Sensoren. Einige Detektoren, die über das gesamte Gesichtsfeld ansprechen. Konturendetektion auf einige e_j.

a6) Räumlicher Gradient (Neigung) von Konturen. Ein e_j für jede Neigung für das Gesichtsfeld.

a7) Zweite räumliche Ableitung (Krümmung) von Konturen. Weiterleitung ähnlich zu a6.

b) Akustisches Feld

b1) Grobes Frequenzraster über den gesamten Bereich mit diskreter Lautstärkeskala.

b2) Feines Frequenzraster über eine Teilzone, das über den gesamten Frequenzbereich verschoben werden kann.

b3) Zeitliche Ableitung der Frequenz.

b4) Zeitliche Ableitung der Lautstärke.

Die Anzahl der Sensoren sollte erheblich über der Anzahl der Rasterzonen liegen, so daß jede von ihnen mit mehreren Sensoren besetzt ist. Dadurch wird vor allem die Bildung der Ableitungen leichter und genauer.

Die technische Aufgabe, die damit gestellt ist, ist im Prinzip erfüllbar, deshalb sollen Einzelheiten hier nicht diskutiert werden. Im Abschn. 4.6 wird das Problem in einem vertieften Sinne nochmals aufgegriffen werden.

Als Resultat stehen schließlich auf n Leitungen elektrische Signale "ein" = 1 und "aus" = O an. Die Leitungen sind entweder direkt den Rasterpunkten oder aber den abgeleiteten Funktionen zugeordnet. Damit sind 2^n verschiedene Kombinationen $(e_j)_i$ möglich. Regt ein entsprechendes Muster die Sensoren an, so sei $(e_j)_i$ "aufgerufen", was auch geschrieben wird als $(e_j)_i = 1$. Dies sei ein Kodiertes Muster.

3. Entscheidungsfunktionen für die Mustererkennung

3.1 Allgemeiner Lösungsweg

Die Vektoren $(e_j)_i$ sind nun bestimmten Begriffen zuzuordnen und umgekehrt. In der Literatur [7] werden die $(e_j)_i$ als Punkte eines n-dimensionalen Hyperwürfels aufgefaßt. Jedem Muster mit einer gewissen Schwankungsbreite entspricht ein Punkthaufen (Cluster) in einem bestimmten Bereich des Hyperwürfels. Es sind dann Hyperflächen als Entscheidungsfunktionen zu konstruieren, die die einzelnen Cluster voneinander trennen. Alle Punkte eines durch die Hyperflächen abgegrenzten Bereiches werden als ein Begriff, also z.B. als a, b usw. interpretiert. Es sind zahlreiche Algorithmen zur Konstruktion und Korrektur der Hyperflächen durch

Lernvorgänge entwickelt worden. Einer der ersten ist das "Perceptron" von Rosenblatt [11]. In den meisten Fällen sind die zu klassifizierenden Muster Schriftzeichen.

Die Zuordnung zwischen Muster und Begriff erfolgt üblicherweise durch den Menschen, der der Maschine in jedem Falle sagt, was das gezeigte Muster bedeutet. Diese jeweils gegebenere Instruktion kann ersetzt werden durch allgemeine Regeln über die Häufigkeit bestimmter Buchstaben und Buchstabengruppen und ermöglicht dann ein Lernen der Maschine auch ohne unmittelbaren Eingriff eines menschlichen Instruktors. So sind selbst Kryptogramme zu entschlüsseln [13]. Dennoch ist die Mitwirkung des Menschen damit nicht eliminiert, seine dauernde Teilnahme ist lediglich ersetzt worden durch eine vorab gelieferte Problemanalyse. Der Prozeß ist darum so gut wie die Analyse und die Fähigkeit des Analysators, das Vorhandene richtig zu erfassen und das zu Erwartende vorherzusehen.

Für das hier verfolgte Konzept ist das Finden der begrifflichen Bedeutung eines Musters zu verallgemeinern und von menschlichem Eingriff unabhängig zu machen.

Allgemein ist ein Muster dann relevant, wenn es entweder das Ergebnis einer Aktion des Systems ist oder wenn es eine Aktion (bzw. Reaktion) des Systems erfordert. An die Stelle der Zuordnung der Muster und Bedeutung durch aufgeprägte Vorschriften soll die Ablaufkette Muster-Aktion-Muster mit den Spuren treten, die sie hinterläßt.

Im Anfangszustand ist das System "leer". Auf den Sensoren der Netzhaut oder der Schnecke des Gehörs werden Erregungsmuster erzeugt, doch führen sie zu keinerlei wie auch immer gearteten Aktionen (Reaktionen) des Systems. Das wird als gleichbedeutend damit angesehen, daß die Muster nicht "erkannt" und nicht "wahrgenommen" werden. Dieser "originale" Zustand wird erst dadurch geändert, daß die motorischen Komponenten eine Aktion ausführen und in zeitlichem Zusammenhang

damit zeitlich veränderliche Muster, also Bewegungen nach Abschn. 2, a4) bzw. b3) und unterschiedliche Muster vor und nach der Aktion auftreten. Der Ablauf "Muster vor der Aktion - Aktion und begleitende bewegte Muster - Muster nach der Aktion" wird im Speicher aufgezeichnet. Damit aber sind die beteiligten Muster auf die Stufe der Wahrnehmung gehoben worden. Tritt das "Muster vor der Aktion" wieder am Sensor auf, so löst es diesmal die betreffende Aktion aus, der Ablauf "Muster - Aktion - Muster" wird erneut registriert und mit der früheren Aufzeichnung verglichen und ggf. verstärkt. Das wiederholt sich und mit dieser zunächst noch primitiven und reflexhaften Reaktion ist die erste Stufe der "Wahrnehmung" gelernt worden.

Im allgemeinen Fall erfolgt dieser Ablauf nur in einer begrenzten Anzahl von Wiederholungen und so lange, bis die beiden durch die Aktion verknüpften Muster regelmäßig die gleichen bleiben. Auf diese Weise lernt das System nach und nach verschiedene Muster wahrzunehmen, d.h. es lernt sie als Anfang, Begleitung und Ende seiner Aktionen und es verhält sich, als seien sie deren Anlaß und Resultat.

Ganz analoge Beobachtungen berichtet Piaget [1] von Kleinkindern. Sie versuchen Aktionen, die zunächst zufällig erzeugt sein können und die zu bestimmten sensorischen Reizen führen, "systematisch" zu wiederholen, bis das Ganze bekannt und "langweilig" geworden ist. Piaget nennt diese repetitiven Abläufe zirkulare Aktionen und gibt ihnen eine bedeutsame Rolle in der Formung der Wahrnehmung. Phillips [13] formuliert das in seiner Darstellung der Arbeiten Piagets, indem er für diese Entwicklungsphase das "the world is my action upon it" zum übergeordneten Prinzip erklärt. Auf diese Weise formt sich das Bild, das sich das Subjekt von der Umwelt macht so und gerade so, wie es für das Handeln in ihr erfordert wird. Die Wiederholung "neuer" Abläufe aber erfüllt die biologische Funktion, Zufallserscheinungen zu eliminieren.

Das Prinzip, Muster durch ihre Veränderungen zur ersten Wahrnehmung zu bringen, gibt Komponenten wie Abschn. 2, a4) und b3), die unmittelbar zur Veränderung gehören, ein besonderes Gewicht. Allgemein bei höheren Tieren und speziell beim Menschen zeigen in der Tat viele Beobachtungen, wie gerade diese "Bewegungsmuster" den Organismus zur unmittelbaren Reaktion veranlassen, sei es durch Reflexbewegungen, sei es durch Adrenalinausschüttungen. Viele dieser Reflexe sind a priori fest einprogrammiert, und bekanntlich nimmt auch der Mensch Bewegungen leicht wahr, während dieselben Objekte in Ruhe leicht übersehen werden. Versuche an Kleinkindern haben z.B. gezeigt, daß gerade auch Veränderungen im Gesichtsfeld, wie z.B. das Wechseln farbiger Lichtbilder, hervorgerufen durch Saugen mit den Lippen an einem entsprechenden Steuermechanismus, schon sehr früh zu einer Rückkopplung und einem intensiven zirkularen Ablauf führen.

Mit den Elementen

- Aktion
- Bewegungsmuster
- Muster vor der Aktion
- Muster nach der Aktion
- begrenzte Zahl zirkularer Wiederholungen

soll die bis hierhin allgemein formulierte Aufgabe der Bildung der Wahrnehmung gelöst werden. Gelingt dies, so ist damit ein System zur Mustererkennung entstanden; denn ein System hat ein Muster dann erkannt, wenn es in eindeutiger Weise darauf reagiert.

3.2 Ein Algorithmus zur Formierung relevanter Muster

Abbildung 1 zeigt das Fließbild eines geeigneten Programms. Eine Aktion A_{ir} bewirkt eine Veränderung in der Umwelt und erzeugt ein Signal A. Das Umweltmuster wird in der bekannten Weise in den

Vektor (e_j) umgewandelt. Die Komponenten . . . , e_m, e_n seien im Sinne von Abschn. 2, a4) zeitliche Ableitungen. Sobald sie auftreten, öffnen sie zusammen mit A das "und"-Gatter 1. (e_j) wird über ein Gatter mit einem von A ausgelösten Taktgeber in periodische Momentbilder zerlegt, die in einer Verzögerungsstrecke verzögert werden. Jede Komponente $e_j = 1$ öffnet zusammen mit dem Impuls aus 1 ein "und"-Gatter 3. Wegen der Verzögerungsstrecke entspricht dies $e_j = 1$ dem Zustand von (e_j) vor dem Einsetzen der Veränderung. Dies sei als $(e_j)_i$ in 5 gespeichert. Wenn die Veränderung aufhört, wird mit $e_m = e_n = O$ über das "nicht"-Gatter das "und"-Gatter 2 geöffnet und nach Verzögerung das "und"-Gatter 4 für jedes jetzt anstehende $e_j = 1$ geöffnet. Das Resultat werde als $(e_j)_r$ in 6 gespeichert.

$(e_j)_i$ und $(e_j)_r$ können als Adressen der Plätze F_i und F_r in einem Speicher F gelesen werden. An die Stelle F_i wird geschrieben ${}_AM_{ir}$, F_r und an die Stelle F_r entsprechend F_i. F_i, F_r und ${}_AM_{ir}$ sind dann Pointer für die entsprechenden Stellen im Speicher F bzw. für den Auslöser der Aktion A. Ausgeschrieben lauten die Pointer bzw. Adressen für F_i und F_r natürlich gerade $(e_j)_i$ und $(e_j)_r$. Bei ${}_AM_{ir}$ werden ggf. noch Zahlenwerte der Bewegungskomponenten e_m, e_n verzeichnet, die eine genauere Steurung der Aktion ermöglichen.

Die Einübung der Mustererkennung erfolgt nach Abb. 2. Bei Auftreten von $(e_j)_i$ und Anwahl von F_i wird die dort stehende Information ${}_AM_{ir}$, F_r gelesen. Außerdem steht dort noch der Zahlenwert einer Funktion $w(r) \leq 1$. Für $w(r) < 1$ wird die Aktion A ausgelöst und das Resultat $(e_j)_x \longrightarrow F_x$ mit dem Pointer F_r verglichen. Wenn $F_x = F_r$, so wird $w(r)$ in F_i um einen vorgegebenen Betrag erhöht und der gesamte Ablauf kann solange wiederholt werden, bis $w(r) = 1$. Wenn $F_x \neq F_r$, wird erneut der Ablauf der Abb. 1 ausgelöst und führt zu anderen F_k, F_s und ${}_AM_{ks}$, die dann ihrerseits eingeübt werden können. Im Falle des Mißerfolgs $F_x \neq F_r$ wird $w(r)$ erniedrigt.

3.3 Struktur des Aspektspeichers

Die beschriebenen Programme ordnen jedem relevanten Vektor $(e_j)_i$ einen Speicherplatz F_i zu, in den als Pointer die ${}_AM_{ir}$ und F_r sowie der Zahlenwert der Funktion w(r) geschrieben werden. Der Inhalt von F_i werde, da er von dem Anfangsmuster $(e_j)_i$ ausgelöst wird, als Vorwärtsschema bezeichnet. Ebenso ist schon ein in F_r zu speicherndes Rückwärtsschema F_i beschrieben worden. Die Schemata sind ihrer Funktion nach denen von Becker [6] ähnlich, wenn auch der Formalismus ein anderer ist. Vor allem ist aber ein Algorithmus definiert, der die Abgrenzung des Schemas und seiner Bestandteile aus der Fülle der ankommenden Muster erlaubt. Die Funktion w(r) entspricht dem von Becker verwendeten confidence-Parameter.

Das Problem der Entscheidungsfunktion für die Mustererkennung wird demnach so gelöst, daß jedem der 2^n Eckpunkte des n-dimensionalen Würfels ein Speicherplatz in F zugeordnet wird, an dem Pointer stehen, falls das Muster relevant ist. Die entsprechenden F_i seien Aspekte genannt. In Abschn. 4.4 wird noch einmal (Fußnote) behandelt werden, wie das Problem der großen Zahlen (2^n) der erforderlichen Speicherplätze umgangen werden kann.

Es muß nun ein Formalismus gefunden werden, der die Clusterbildung, d.h. die Festlegung und Erkennung von Ähnlichkeiten erlaubt. Dazu wird der Speicher F (Aspektspeicher) nach dem Index p der Adressen $(e_j)_i$, d.h. nach der Zahl der Komponenten $e_j = 1$ geordnet. Als "unterste Schicht" seien alle Adressen mit p = 1 bezeichnet, als nächste p = 2 usw. bis p = n. Die $(e_j)_i$ seien so organisiert (s. Abschn. 4.6), daß die Komponenten $e_j=1$ die für die Erkennung wesentliche Information tragen. Dann enthalten die Aspekte mit steigendem p mehr zusätzliche "Merkmale". Sie werden dadurch differenzierter. Mit abnehmendem p werden sie umgekehrt genereller.

Für den Erkennungs- und Zuordnungsprozeß sollen dann die folgenden Regeln gelten:

1. Bei Auftreten eines $(e_j)_i$ werden alle mit Pointern besetzten Aspekte aufgerufen (angewählt), deren Adressen mindestens eine 1 an einer

Stelle haben, wo auch $(e_j)_i$ eine 1 hat und die keine 1 an Stellen haben, wo $(e_j)_i$ eine 0 hat.

2. Im Zuge der Lern- und Einübungsprozesse nach Abb. 1 und 2 werden die Pointer in ein F_i mit so kleinem p geschrieben, daß eine eindeutige Zuordnung gerade noch möglich ist.

3. Bei einem Mißerfolg $(e_j)_x \neq (e_j)_r$ der Klassifikation im Sinne der Abb. 2 werden für die F_i und F_r nach dem Schema der Abb. 1 Plätze mit höherem p gewählt (evtl. Sperre für die früheren Plätze). Dieser Prozeß wirkt im Sinne einer Differenzierung.

4. Stehen als Folge der Prozesse nach Abb. 1 und 2 die gleichen Pointer in mehreren F_i, so werden sie an eine neue Adresse übertragen, die nur dort eine 1 enthält, wo jedes der bisherigen F_i auch eine 1 in seiner Adresse besitzt. Das neue p ist dann kleiner als die alten. Dieser Prozeß wirkt als Generalisierung.

Die zur Ausführung dieser allgemeinen Aufgaben erforderlichen Operationen werden im folgenden genauer beschrieben werden.

Mit diesem Organisationsprinzip wird erreicht, daß Gruppen von Mustern, die einander ähnlich sind und durch eine bestimmte Aktion in gleicher Weise transformiert werden, durch ein einziges, möglichst einfaches Muster repräsentiert werden. Der den entsprechenden Punktcluster im Hyperwürfel repräsentierende Eckpunkt ist dann derjenige, der dem Ursprung jeweils am nächsten liegt.

Ein Muster wird erst dann als anders als ein bereits bekanntes erkannt, wenn es durch Anwendung der gleichen Aktion anders transformiert wird. Die Pointer ${}_AM_{ir}$, F_r steuern die Transformationen des Musters $(e_j)_i$ durch die in der Außenwelt stattfindende Aktion A. ${}_AM_{ir}$ kann als Operator aufgefaßt werden, der den äußeren Vorgang im Systeminneren

nachbildet und den der Mustertransformation entsprechenden Übergang von einer Position F_i zu einer Position F_r vollzieht. Die F_i, F_r haben die Funktion von Bildern oder Symbolen der äußeren Objekte. Die Objekte sollen sich für das System nur dann unterscheiden, wenn sie sich gegenüber seinen Aktionen verschieden verhalten. Das äußert sich im System durch die Transformationseigenschaften der Symbole F_i, F_r. Dadurch und durch die beschriebenen Prozesse der Differenzierung und Generalisierung nach dem Index p wird erreicht, daß ohne Einwirkung eines menschlichen Instruktors für die Abgrenzung der Aspekte nur genau so viel Komponenten (e_j) und nicht mehr verwendet werden, als zur eindeutigen Klassifizierung unbedingt erforderlich sind. Redundante Komponenten, die nicht mit einem unterschiedlichen Verhalten des Musters verbunden sind, werden abgebaut und nicht wahrgenommen. Ein solches System unterscheidet zum Beispiel nicht zwischen roten und grünen Objekten, die sonst übereinstimmen, solange die Farbkomponente nicht irgendwo mit einem unterschiedlichen Verhalten gegenüber Aktionen verknüpft ist. Für die weitere Darstellung sollen zunächst einige formale Operationen definiert werden. x)

3.4 Definition einiger formaler Operationen

Die Aspekte F_i entsprechen n-stelligen Binärzahlen, die ein Code $(e_j)_i$ für das zugehörige Muster sind und nach aufsteigendem Index p geordnet werden. Innerhalb des so gegebenen Feldes können verschiedene Operationen definiert werden:

Operation 1: Der zwei Aspekten F_i und F_k gemeinsame Aspekt F_m ergibt sich analog zu den Operationen der Mengenlehre als Durchschnitt

(1) $$F_m = F_i \cap F_k$$

Dazu werden die zu den F_i, F_k gehörigen Binärzahlen bzw. Vektoren (e_j) komponentenweise vereinigt nach den Regeln:

x) Anmerkung: Das Prinzip der eineindeutigen Zuordnung zwischen Aspekt (Muster) und Transformationseigenschaften mißlingt im wesentlichen dann, wenn nicht genügend sensorische Musterkomponenten zur Verfügung stehen oder wenn die Kodierung unzweckmäßig ist. (... weiter S. 17 Fußnote ..)

$$1 \cap 1 = 1$$

$$1 \cap 0 = 0 \cap 1 = 0 \cap 0 = 0$$

Die sich ergebende Binärzahl wird als neue Adresse F_m gelesen. p wird bei dieser Operation im allgemeinen erniedrigt.

Operation 2: Die Vereinigung zweier Aspekte ergibt einen Aspekt mit höherem p nach der Formel

(2) $$F_n = F_i \cup F_k$$

Auch diese Operation wird kompenentenweise an den zugehörigen Binärzahlen ausgeführt nach der Regel:

$$1 \cup 1 = 1 \cup 0 = 0 \cup 1 = 1$$

$$0 \cup 0 = 0$$

Operation 3: Ein Ternärterm T_{ir} wird durch die Operation

(3) $$T_{ir} = F_r - F_i$$

gebildet. Hier werden die zu den beiden F gehörenden Binärtermen komponentenweise voneinander abgezogen nach der Regel

$$0 - 0 = 1 - 1 = 0$$

$$1 - 0 = 1$$

$$0 - 1 = -1$$

T_{ir} ist dann aus den Ziffern 0, 1 und (- 1) n-stellig aufgebaut. Für den Sonderfall $F_r = F_i \cap F_k$ ergibt sich nach (3) $T_{ir} = F_r - F_i = F_k$; d. h. eine Komponente mit (- 1) tritt nicht auf.

x) (. . Fortsetzung Fußnote v. S. 16)
Die Kodierung kann nach Abschn. 4. 6 verbessert werden. Bei Mißlingen der Zuordnung wird der Automat das betreffende Schema mit mehrdeutigen Operatoren entweder nicht ausführen oder dem Operator mit der größten Erfolgshäufigkeit folgen.

<u>Operation 4:</u> T_{ir} kann als Operator aufgefaßt und auf F_i angewandt werden nach der Regel

(4) $$F_r = T_{ir} + F_i$$

mit komponentenweiser Anwendung

$$1 + O = O + 1 = 1$$
$$(-1) + 1 = O$$
$$O + O = O$$

Die Operationen

$(-1) + O$ und $1 + 1$ seien nicht ausführbar.

Kommen sie vor, so sei T_{ir} auf F_i <u>nicht anwendbar</u>.

<u>Operation 5:</u> Aus T_{ir} und F_k kann ein neuer Aspekt F_m gewonnen werden, dessen Adresse nur dort eine 1 hat, wo sowohl M_{ir} als auch F_k keine O haben. Der Index wird dadurch ggf. reduziert.

(5) $$F_m = F_k \cap T_{ir} \qquad p(m) \leq p(k)$$

Komponentenweise Ausführung nach der Regel:

$$1 \cap (-1) = 1 \cap 1 = 1$$
$$O \cap (-1) = O \cap 1 = O \cap O = 1 \cap O = O$$

Operation 6: Zwei Operatoren T_{ir} und T_{pq} können zu einem Operator T_{xy} vereinigt werden nach

(6) $$T_{xy} = T_{ir} \cup T_{pq}$$

Komponentenweise Ausführung nach der Regel

$$1 \cup O = O \cup 1 = 1 \cup 1 = 1$$

$$(-1) \cup (-1) = O \cup (-1) = (-1) \cup O = (-1)$$

$$1 \cup (-1) = (-1) \cup 1 = O \cup O = O$$

Insbesondere läßt sich ein Operator T_{ri} definieren, der die Bedingung

$$T_{ir} \cup T_{ri} = O$$

erfüllt. Die Null auf der rechten Seite bezeichnet einen Ternärterm, dessen sämtliche Komponenten Null sind. Aus der dritten Zeile der Rechenregeln folgt, daß beim Übergang von T_{ir} auf T_{ri} immer gerade (- 1) gegen 1 und umgekehrt ausgetauscht wird, während die Komponenten O bleiben. Mit Operation 4 gilt

$$F_r = T_{ir} + F_i \quad \text{und} \quad F_i = T_{ri} + F_r$$

T_{ri} <u>annulliert</u> T_{ri} und umgekehrt.

Noch verschiedene andere Operationen wären formal möglich, doch da sie im folgenden nicht benötigt werden, soll auf sie verzichtet werden.

3.5 Operativer Aufbau des Aspektfeldes

Die allgemeinen Ausführungen zum Aspektspeicher in Abschn. 3.3 und die formalen Operationen des Abschn. 3.4 sollen jetzt benutzt werden, das Aspektfeld genauer zu strukturieren. In Abb. 3 sind einige Aspekte mit den eingetragenen Pointern dargestellt. Die Richtung des Pfeiles bezeichnet das Transformationsverhalten, also das bei Ausführung der Aktion erzielte Ergebnis F_r oder F_a usw. Der Großbuchstabe bezeichnet die spezielle Aktion in der Umwelt, hier A bzw. B.

F_1 werde bei Ausführung von A nach F_a bei Ausführung von B nach F_b transformiert. Die Adressen sind entsprechend $(e_j)_a$ und $(e_j)_b$.
Also muß im Speicher bei F_1 stehen:

$$F_1 \left[{}_A M_{1a}, F_a, \ldots \right]$$

und

$$F_1 \left[{}_B M_{1b}, F_b, \ldots \right]$$

Der vor den Operator gesetzte Index A bzw. B gibt die Zuordnung zwischen diesem und der Adresse des Aktuators.

Die Punkte weisen auf weitere Eintragungen, wie z.B. die in Abschn. 3.2 erwähnte Funktion w(a), w(b) hin.

Bei dem zu F_2 gehörigen Muster führt die Aktion A zum gleichen Resultat wie bei F_1, die Aktion B dagegen zu einem unterschiedlichen.

Geschrieben:

$$F_2 \left[{}_A M_{2A}, F_a, \ldots \right]$$

$$F_2 \left[{}_B M_{2c}, F_c, \ldots \right]$$

Bei F_3 führt umgekehrt B zum gleichen, A zu einem anderen Resultat als bei F_1.

$$F_3 \left[{}_A M_{3d}, F_d, \ldots \right]$$

$$F_3 \left[{}_B M_{3b}, F_b, \ldots \right]$$

Bei F_4 besteht kein Unterschied zu F_2

$$F_4 \left[{}_A M_{4a}, F_a, \ldots \right]$$

$$F_4 \left[{}_B M_{4c}, F_c, \ldots \right]$$

und bei F_5 liefern A und B das gleiche Resultat:

$$F_5 \left[{}_A M_{5a}, F_a, \ldots \right]$$

$$F_5 \left[{}_B M_{5a}, F_a, \ldots \right]$$

Im Hinblick auf die Aktion A bilden F_1, F_2, F_4 und F_5 einen Cluster. Es wird nun postuliert, daß zwischen dem Muster, das das Objekt darbietet und seinem Transformationsverhalten eine eindeutige Relation besteht. Die Lebewesen benutzen ihre Sinnesorgane unter dieser Voraussetzung. Dann müssen F_1, F_2, F_4, F_5 bestimmte gemeinsame Komponenten besitzen und diese sind die wesentlichen für das Verhalten gegenüber A.

Ein großenteils oder vielleicht nur diese Komponenten enthaltender <u>Oberaspekt</u> F_o kann erzeugt werden nach

$$F_o' = F_1 \frown F_2 \qquad F_o'' = F_1 \frown F_4 \qquad F_o''' = F_1 \frown F_5$$

$$F_o'''' = F_o' \frown F_o''$$

$$F_o = F_o'''' \frown F_o'''$$

F_o ist der Durchschnitt aus den Aspekten mit demjenigen unter ihnen, der jeweils das kleinste p besitzt. Es kann beispielsweise auch $F_o = F_1$ gelten. Im Zuge der Generalisierung wird eingetragen

$$F_o \left[{}^{A}M_{oa}, F_a, \ldots \right]$$

F_3 dagegen muß mindestens eine Komponente enthalten, die in F_o nicht vorkommt und die für das andersartige Verhalten gegenüber A charakteristisch ist. Es möge gelten

$$F_3 = F_o \smile F_3'$$

wo F_3' das für das neue Verhalten wesentliche Teilmuster ist. (Natürlich kann vorher F_3 durch Durchschnittsbildung mit anderen Aspekten gleicher Transformationseigenschaften ebenfalls auf die wesentlichen Komponenten reduziert werden.) F_3 ist dann ein (differenzierterer) <u>Unteraspekt</u> von F_o.

Dieser Sachverhalt impliziert, daß außerdem gilt

$$F_o = F_3 \frown F_o$$

Wenn also ein Muster ansteht, das sowohl die Adressen von F_o und von F_3 enthält, kann die Anwahl von F_o und das Lesen der dort stehenden Pointer offensichtlich nicht die richtige Reaktion des Systems im Erkennungsprozeß sein. Man kann das berücksichtigen, indem in F_o eine weitere Eintragung, eine Negation vorgesehen wird, die anzeigt, daß bei gleichzeitigem Auftreten der zu F_3' gehörigen Adresse im Muster F_o nicht angesprochen ist, etwa

$$F_o \left[{}_A M_{oa}, F_a, \ldots, \overline{F_3'}, \ldots \right]$$

wo der Querstrich die Negation bezeichnet.

Gegenüber der Aktion B bilden F_1 und F_3, F_2 und F_4 und schließlich F_5 je einen Cluster. Die Durchschnitte sind F_{oo}, F_{ooo} und F_5.

Offensichtlich sind F_1, F_2, F_3 und F_5 verschiedenen Objekten zuzuordnen, da sie sich zwar gegen eine Aktion teilweise gleich, gegenüber verschiedenen Aktionen aber alle verschieden verhalten. F_2 und F_4 dagegen verhalten sich allgemein gleich.

Es erscheint sinnvoll, in dem in F erzeugten Bild der Umwelt den verschiedenen Objekten entsprechenden Aspekten Namen zuzuordnen. Der F_2 und F_4 gemeinsame Name müßte an der Stelle

$$F_N = F_o \smile F_{ooo}$$

eingetragen werden, wobei der Index

$$p\,(F_N) \leq p\,(F_o \smile F_{ooo})$$

ist. In F_N müßte ein Name N_1 als Pointer stehen:

$$F_N \left[\ldots, N_1, \ldots\right]$$

Die Eintragung von Operatoren, Aspektadressen und Negationen erübrigt sich an sich, da gemäß Abschn. 3. 3, Regel 1 F_o und F_{ooo} mit aufgerufen werden, die die notwendige Information bereits enthalten. Der Name ist ebenfalls ein Pointer, der zu einem die Verbalisierung bewirkenden Aktuator führt; im einfachsten Fall ein Fernschreiber, der ein für den Namen charakteristisches Wort schreibt.

Die Darstellung nach Abb. 3 zeigt, daß die einzelnen Cluster in vielfacher Weise einander durchdringen können, wenn mehrere Aktionen möglich sind. Jeder Cluster wird durch einen mittels des gemeinsamen Durchschnitts definierten Aspekt und dort ggf. eingetragene abgrenzende Negationen beschrieben, und verschiedene Cluster können ihrerseits wieder einen nicht leeren Durchschnitt besitzen. Es war schon gesagt worden, daß die Adresse von F_o im Hyperwürfel den dem Ursprung nächsten Punkt des durch A gleichartig transformierten Clusters bezeichnet. Punkte dieses Clusters können gleichzeitig auch noch zu anderen Clustern gehören und umgekehrt.

Die in den Arbeiten zur Mustererkennung übliche Beschreibung der Ähnlichkeit zweier Muster durch den Abstand der zugehörigen Punkte im Hyperwürfel erweist sich als nicht ausreichend. Ähnlichkeit ist vielmehr in bezug auf bestimmte Aktionen zu definieren. F_2 ist F_1 ähnlich in bezug auf A, F_4 aber ist ihm ähnlich in bezug auf A <u>und</u> B und es läßt sich nicht ohne weiteres sagen, ob der zu F_1 gehörige Punkt im Hyperwürfel weiter von dem für F_2 entfernt ist als dieser von dem für F_4. Dennoch ist die Ähnlichkeit von F_2 zu F_4 größer als zu F_1. Auf die Tatsache, daß jeder Punkt im Hyperwürfel Nachbarn hat und so Ähnlichkeiten in großer Menge möglich sind und ohne ein zusätzlich

gegebenes Gewicht wenig Aussagekraft besitzen, hat auch Watanabe [14] in mehr prinzipieller Form hingewiesen. Die Bildung der Durchschnittsaspekte und die Zahl der verschiedenen Pointer, die diesen gemeinsam sind, erlauben die Definition eines solchen Gewichts.

An dieser Stelle ist ein grundsätzlicher Hinweis angebracht. Die Strukturierung der internen Bilder der Umweltobjekte geschieht nach deren Verhalten. Inhärent setzt dabei das System einen eindeutigen, deterministischen Ablauf des Geschehens gleichsam induktiv voraus. Demzufolge hat jedes Schema eine und nur eine Zusammensetzung, oder es ist eben ein anderes Schema, das ein anderes Verhalten anderer Objekte beschreibt. Die von Becker [6] beschriebene Möglichkeit, bestimmte Bestandteile der Schemata auftreten oder nicht auftreten zu lassen, und den Grad der Auftretenswahrscheinlichkeit durch eine als "criteriality" bezeichnete Größe zu beschreiben, hat hier definitiv und absolut keinen Platz. Dies ist einer der wichtigsten, die Grundstruktur dieses Systems beschreibenden Züge.

3.6 Spezielle Eigenschaften der Operatoren

Die Operatoren ${}_AM_{ir}$ wurden bisher als Pointer für einen bestimmten Aktuator A verstanden, der die Umwelt in dem Sinne verändert, daß das Muster $(e_j)_i$ in das Muster $(e_j)_r$ übergeht. Verallgemeinert transformiert M_{ir} F_i in F_r. Man kann dann unter Benutzung der Operationen 3 und 4

$$ {}_AM_{ir} \equiv T_{ir} $$

setzen.

Diese Darstellung hat zwei interessante Vorteile:

- Für T_{ir} sind im Zusammenhang mit Operation 4 Regeln für die Nichtanwendbarkeit angegeben, die eine zusätzliche Kontrolle der Prozesse ermöglichen.

- Die Operation 5 mit der Durchschnittsbildung zwischen Operator und Aspekt eröffnet einen weiteren Weg zur Generalisierung und damit der Elimination irrelevanter Komponenten.

3.7 Aspekte durch "passive" Veränderungen

In der bisherigen Darstellung der Generierung von Aspekten gemäß Abb. 1 war dazu eine Aktion A des Systems erforderlich. Natürlich gibt es aber auch Veränderungen in den Mustern, die sich ohne eine Aktion des Systems ereignen. Auch hierbei können zunächst nicht "wahrgenommene" Muster durch Definition der Aspekte erkennbar gemacht werden. Es wäre im Prinzip möglich, den Prozeß der Abb. 1 auch ohne die Auslösung durch A stattfinden zu lassen. Nur wäre dann keine Reaktion des Systems auf die Muster anzugeben und der Erkennungsprozeß wäre sinnlos. Wenn dagegen einer der beiden Aspekte, z.B. F_r, schon definiert ist, kann im wesentlichen nach dem gleichen Schema F_i gebildet werden und erhalte die Pointer

$$F_i \left[M_{ir}, F_r, \ldots\right]$$

Eine Angabe der Adresse A beim Operator erübrigt sich. Mit $M_{ir} \equiv T_{ir}$ kann die Zieladresse durch Operation 4 ermittelt werden. Der bereits definierte Aspekt F_r kann insbesondere schon a priori gegeben sein. Beim Vergleich zum lebenden Wesen kann man an Empfindungsmuster wie Hunger, Durst oder Sättigung denken, die auch schon <u>vor</u> allen Lernprozessen Reaktionen auslösen. In analoger Weise lassen sich hier andere Aspekte definieren und damit verbinden. Der Ablauf wäre dann abweichend, aber doch in gewisser Relation zu Abb. 1 und Abb. 2:

- Wahrnehmung eines $(e_j)_i$
- Anwahl F_i
- Lesen M_{ir} F_r

- Auftreten von Veränderungskomponenten e_m, e_n, . . .
- Gleichzeitig Nichterfolg F_r -$(e_j)_r$
- Bildung F_x und M_{ix} als Aspekt nach Verschwinden von e_m, e_n, . . . analog zu Abb. 1
- Generalisierung und Differenzierung.

3.8 Operationen zur Generalisierung und Differenzierung

In Abschn. 3.3 war bereits in allgemeiner Form dargestellt worden, wie durch die Vorgänge der Generalisierung und Differenzierung die Anzahl der "wahren" Komponenten (e_j = 1) in den Aspektadressen oder der Index p des Aspektes gerade so groß und nicht größer gewählt wird, daß eine eindeutige Wahrnehmung des von den Sensoren aufgenommenen Musters möglich ist.

Die Generalisierung besteht darin, daß bei einer erfolgreichen Wahrnehmung, d. h. bei einer Übereinstimmung zwischen tatsächlicher und erwarteter Transformation des Musters gemäß Abb. 2, der Index reduziert werden kann.

Die Differenzierung besteht darin, daß bei einem Mißerfolg in der Wahrnehmung der Index erhöht wird.

Generalisierung und Differenzierung sollen nun in ihrem formalen Ablauf etwas genauer beschrieben werden.

1. Generalisierung

Der Aspektspeicher ist im Anfangszustand leer oder enthält doch nur wenige Eintragungen (a-priori-Aspekte) mit sehr geringem p. Außerdem ist es möglich, die Zahl der Komponenten des Vektors $(e_j)_i$ dadurch zu reduzieren, daß mehrere von ihnen, die der gleichen Musterqualität wie Lage auf dem Raster, Kontur, Steigung, Krümmung, zeitliche Ableitung usw. zugeordnet sind, jeweils zu einer einzigen zusammengefaßt werden

können. Dieser Prozeß heiße Rasterung. Verstärkte Rasterung vermindert die Zahl der Komponenten e_j, verminderte Rasterung erhöht sie.

Es seien e_j, e_{j+1} und e_{j+2} drei Komponenten, die durch die Rasterung zusammengefaßt werden. Das tatsächliche Muster (e_j, e_{j+1}, e_{j+2}) wird bei maximaler Rasterung dann in eines umgesetzt, in dem $e_{j+1} = e_{j+2} = O$ gilt.

Also $$(e_j, e_{j+1}, e_{j+2}) \longrightarrow (e_j', O, O)$$

Es gelte $e_j' = 1$, wenn e_j, e_{j+1} oder e_{j+2} oder mehrere von ihnen 1 sind. Nur für $e_j = e_{j+1} = e_{j+2} = O$ wird auch $e_j' = O$. Das neuerzeugte Muster wird zur Definition der Aspekte benutzt und besitzt weniger "wahre" Komponenten, die entsprechend einen Aspekt mit geringerem Index zur Folge haben.

2. Differenzierung

Im Zusammenhang mit einer Aktion werden nach Abb. 1 zwei Aspektadressen $(e_j)_i$ und $(e_j)_r$ formuliert, in die die Eintragungen

$$F_i \left[{}_AM_{ir}, F_r, w(r), \ldots \right]$$

und

$$F_r \left[F_i, \ldots \right]$$

geschrieben werden.
w(r) hat einen spezifizierten Anfangswert. Die Punkte lassen die Möglichkeit für weitere, später zu definierende Eintragungen.

Für ${}_AM_{ir}$ ist in dem Vorwärtsschema nur die Adresse des Aktuators A relevant. Dort kann ggf. der Operator

$$M_{ir} = F_r - F_i$$ nach Abschn. 3.4, Operation 3 stehen.

3. Generalisierung

Die so gebildeten F_i und F_r können noch viele Komponenten enthalten, die von der eigentlichen Operation gar nicht betroffen werden, sondern zufälligen Hintergrundsmustern zugehören. Darum erfolgt eine Bereinigung mittels Operation 3.4.5

$$F_i^* = F_i \cap M_{ir}$$

und

$$F_r^* = F_r \cap M_{ir}$$

Die F_i^* und F_r^* treten an die Stelle der F_i und F_r und erhalten die dort stehenden Eintragungen, die dort gelöscht werden.

4. Differenzierung

Nach Abb. 2 tritt mit $F_x \not\vdash F_r$ Mißerfolg der Wahrnehmung auf. Mißerfolg bedeutet dabei

$$p(F_x) - p(F_x \cap F_r) > \varepsilon$$

mit einem vorher definierten ε.
Zunächst kann dann nochmals die Operation nach Abb. 1 angewandt werden. Dadurch kann auf jeden Fall das neue F_r definiert werden. Wenn jedoch das dabei gebildete F_i nicht von dem vorher vorhandenen, andere Transformationseigenschaften besitzenden F_i abweicht, muß eine andere Differenzierungsoperation für F_i angewandt werden. Als erstes kommt eine Verminderung der Rasterung in Frage.

5. Generalisierung

In gewissen Zeitabständen wird der Aspektspeicher daraufhin überprüft, ob bei verschiedenen Aspekten F_i, F_k usw. der gleiche Zielaspekt F_r

für die gleiche Aktion A eingetragen ist. Das Auffinden der so definierten F_i, F_k wird ohne großen Aufwand durch die Existenz der Rückwärtsschemata F_r $[F_i, \ldots]$, usw. ermöglicht.
Dann wird gebildet

$$F_m = F_i \cap F_k$$

und

$$F_m \left[{}_A M_{mr}, F_r, \ldots\right]$$

6. Differenzierung

Schließlich können auch bei aufgehobener Rasterung Wahrnehmungsprozesse zum Mißerfolg führen. Im nächsten Schritt kann dann das Muster auf das Vorhandensein zusätzlicher weiterer Muster abgesucht werden, die ihrerseits zu bereits definierten Aspekten gehören. Sei F_i der nicht zum Erfolg führende Aspekt und F_k ein weiterer, so wird das spezielle Ergebnis der Aktion A in

$F_m = F_i \cup F_k$ versuchsweise eingetragen.

Es bedarf jedoch einer größeren Anzahl zyklischer Wiederholungen (langsames Wachsen von w(r) zum Grenzwert), bis im weiteren Zusammenwirken von Generalisierung und Differenzierung F_m eindeutig definiert wird.

7. Differenzierung

Ein ebenfalls hierher gehörender Differenzierungsprozeß ist das Eintragen von Negationen in bestehende Aspekte. Da nach Generalisierung 5 der den Aspekt charakterisierende Pointer in die Adresse mit dem niedrigsten Index eingetragen wird, müssen dort spezifische Abgrenzungen des durch gleiche Transformationseigenschaften definierten Clusters stehen, die als Negationen geschrieben werden können.

F_o und F_3 (Abb. 3) seien zwei Aspekte mit unterschiedlichen Transformationsverhalten gegenüber A, die auf die bisher beschriebene Weise

definiert worden sind. In gewissen Zeitabständen wird der Aspektspeicher daraufhin überprüft, ob Aspekte F_3 und F_o existieren, die die Form

$$F_o \left[{}_A M_{oa}, F_a, \ldots \right]$$

und

$$F_3 \left[{}_A M_{3d}, F_d, \ldots \right]$$

mit

$$F_a \neq F_d \quad (\text{bzw. } p(F_a) - p(F_a \cap F_d) > \varepsilon)$$

haben und für die gilt

$$F_o \cap F_3 = F_o .$$

Das Auffinden der F_o, F_3 wird erleichtert, wenn an der Adresse A die beiden Operatoren ${}_A M_{oa}$ und ${}_A M_{3d}$ verzeichnet sind.

Gibt es zwei derartige F_o, F_3, so wird gebildet

$$F_3' = F_3 - F_o$$

und

$$F_o \left[{}_A M_{oa}, F_a, \ldots, \overline{F_3'}, \ldots \right] .$$

Abschlußbemerkung zum Prozeßablauf:

Mit diesem Ineinander und Gegeneinander von Generalisierung und Differenzierung werden die Aspekte nach und nach mit den richtigen Komponenten gebildet. Die weiterführenden Pointer werden immer nur in denjenigen Aspekt eingetragen, der von den zur gleichen Transformationseigenschaft gehörenden Cluster den geringsten Index besitzt und dessen Komponenten $e_j = 1$ auch in allen anderen Adressen des Clusters als 1 enthalten sind.

Beim Auftreten eines Musters an den Sensoren werden alle Aspekte angewählt, deren Adressen in bezug auf Komponenten $e_j = 1$ als Untermuster im Muster enthalten sind. Entspricht F_i genau dem auftretenden Muster, so werden also alle F_k, F_e, F_m usw. mit angewählt, die die Bedingung

$$F_k \cup F_l \cup F_m \cup \ldots = F_i \qquad \text{erfüllen.}$$

4. Heuristische Prozesse

4.1 Suchprozesse im sensorimotorischen Bereich

Die Aspekte mit ihren Pointern bilden den Ablauf von Ereignissen der Umwelt in das Systeminnere ab. Man kann sie als Vorwärtsschemata bezeichnen, bei denen ein Muster eine bestimmte Aktion auslöst, die zu einem neuen Muster führt. Der Ablauf ist einem Reflex vergleichbar, der durch die Größe der Funktion w(r) kontrolliert wird.

Ebenso wichtig für ein sensorimotorisches System, das Ziele verfolgen soll, ist aber der umgekehrte Ablauf, bei dem ein Zielaspekt gegeben ist, der durch Aktionen des Systems aus irgeneinem Anfangszustand zu realisieren ist. Dazu werden Such- oder heuristische Prozesse benötigt.

Heuristische Algorithmen sind für die Lösung bestimmter logischer Probleme, beispielsweise für das Schachspiel entwickelt worden [3]. Gegeben sind dabei immer ein genau definierter Anfangszustand, ein genau definierter Endzustand (Ziel oder top-goal) und eine endliche Anzahl von Operationen, die in der richtigen Reihenfolge angewandt den Anfangs- in den Endzustand transformieren. Das Hauptproblem ist also die Reihenfolge der Operationen.

Solche Prozesse lassen sich als Baumstrukturen darstellen, wie dies Abb. 4 zeigt. Die Knoten bedeuten die Zustände, die Linien, die Operationen, die sie ineinander überführen. Der Anfangszustand ist G_1. Nun werde zuerst A ausgeführt, dann D und G_5 als aussichtslos für die Erreichung des Ziels erkannt. Nun wird B an G_1 versucht, und nach abermaligem Mißerfolg führen schließlich C und dann B zum gewünschten Ziel G_7. Der "Erfolg" und "Mißerfolg" der einzelnen Operationen kann durch einen geeignet definierten heuristischen "Abstand" vom Ziel gemessen werden, der nach jedem Schritt bestimmt wird.

Es sind viele Strategien entwickelt worden, durch die die Reihenfolge der Operationen so vorgeplant wird, daß möglichst wenig blindes Probieren erforderlich ist und eine geringe Schrittzahl zum Ziele führt. Dabei können insbesondere Erfahrungen aus früheren Prozessen benutzt werden. Die von Becker [6] zitierte und benutzte "progressive deepening strategy" [15] ordnet jeder Operation Zahlenwerte zu, die nach einem bestimmten Formalismus aus ihrer erfahrenen Zuverlässigkeit und ihrem Aufwand gebildet und von Schritt zu Schritt reduziert werden. Jeweils die Operation, bei der dieser Zahlenwert am höchsten ist, wird als nächste ausgeführt.

Man kann den heuristischen Baum der Abb. 4 auch in umgekehrter Richtung interpretieren und G_7 als Anfangszustand, G_1 dafür als Ziel auffassen. Der von G_1 ausgehende Suchprozeß ermittelt dann rückwärts, mit der letzten Operation beginnend, wieder die zwischen G_7 und G_1 transformierende Operatorenfolge, wobei auch hier ein heuristischer Abstand zur Bewertung der Zwischenresultate dient. Die Grundaufgabe, die Ermittlung der Reihenfolge der Operationen, hat sich gegenüber dem Vorwärtsablauf grundsätzlich nicht geändert.

Interessanterweise stellt man fest, daß die Ausgangslage für heuristische Prozesse auf der sensorimotorischen Ebene doch wesentlich anders ist, als für logische Prozesse, durch die Integrale ausgewertet, Sätze der Aussagenlogik bewiesen werden usw. Statt einer definierten Ausgangssituation liegt hier eine im hohen Maß unklare Ausgangssituation vor. Statt der Frage, welche Operationen führen vom Zustand 1 in den Zustand 2 ? lautet die Aufgabe: Suche den in 2 transformierbaren Zustand 1, der sich in der Umwelt möglichst weitgehend wahrnehmen läßt. Der Grund dafür ist die schon erwähnte große Belastung der natürlichen Musterinformation mit Irrelevanten. Oder um an eine von Newell und Ernst [5] durchgeführte GRS-Fallstudie über einen Affen anzuknüpfen, der einige Kisten aufeinanderstellen und einen Stock ergreifen muß,

um eine oben im Käfig aufgehängte Banane zu erreichen: Das größere Problem des Nahrung suchenden Affen besteht im Gegensatz dazu im allgemeinen darin, in einer ungeheuer komplexen Umwelt möglichst rasch die unsichtbar irgendwo in den Bäumen hängenden Früchte zu finden und dann mit gut eingeübten Manipulationen zu pflücken, nicht aber stattdessen die deutlich sichtbare Banane mit sinnvoll aufeinander folgenden Manipulationen zu erreichen. Die letztere Aufgabe erfordert vielleicht mehr Intelligenz; doch im sensorimotorischen Bereich müssen erst einmal die geeigneten Anfangsmuster in der Umwelt gefunden werden.
Angewandt auf die Baumstruktur der Abb. 4, die als Rückwärtsablauf mit G_1 als Ziel gelesen wird, bedeutet dies:

Statt aus den Operatoren A, B, C, D die zu suchen, die G_7 nach G_1 überführen, ist bei festliegenden A bis D und gegebenen Kombinationen aus diesen unter einer Zahl von Zuständen G_5, G_6, G_7, G_8 derjenige Zustand G_7 zu suchen, der der vorliegenden Umweltsituation am besten entspricht.

Die Zustände G_1 usw. werden nun mit den Aspekten F_i identifiziert, und der Suchprozeß ist dann abgeschlossen, wenn für einen Zielaspekt F_r ein Aspekt F_i $[M_{ir}, F_r, \ldots]$ gefunden wird, für den von den Sensoren her ein Muster $(e_j)_i$ ansteht. Dann ist eine Aktion A möglich, die das Ziel, das Muster $(e_j)_r$, tatsächlich zu erreichen erlaubt.

Abb. 5 zeigt schematisch den Ablauf des Suchprozesses. F_r ist der Zielaspekt. Im Speicher gebe es drei von F_r aus über die Rückwärtsschemata erreichbare Aspekte F_e, F_m, F_n, für die Vorwärtsschemata, die auf F_r führen, definiert sind. Für alle wird abgefragt, ob ein entsprechendes Muster ansteht. Ist dies nicht der Fall, kann der Prozeß mit F_e, F_m oder F_n als Zielaspekt wiederholt werden und liefere schließlich F_g, dem ein $(e_j)_g$ in der Umwelt gegeben ist. Damit wird eine Folge von zwei

Aktionen definiert, die zum Ziel führen. Auch hier zeigt sich der Suchprozeß als Baumstruktur. Auch hier ist eine Strategie erforderlich, die das Auffinden von F_g möglichst beschleunigt.

Um von F_r aus die Adressen von F_e usw. zu finden, benötigt man Rückwärtsschemata F_r $[F_e, \ldots]$ neben den bisher betrachteten Vorwärtsschemata. Es ist kein prinzipielles Problem, beim Definieren von F_e durch den Prozeß der Abb. 1 auch in F_r eine entsprechende, auf F_e zurückverweisende Eintragung zu machen.

Für die Festlegung der Reihenfolge der einzelnen Schritte kann zunächst ebenfalls eine "progressive deepening strategy" angewandt werden. Den Transformationen $F_e \longrightarrow F_r$, $F_c \longrightarrow F_e$ usw. können je nach Häufigkeit ihrer Benutzung Gewichte zugeordnet werden. Ebenso kann ein Maß für den Aufwand der zugehörigen Aktionen A etwa durch die Zahl von elementaren Einzelaktionen, aus denen A zusammengesetzt ist, definiert werden. Damit lassen sich bei den Knoten des Baumes stehende Zahlenparameter bilden, durch die die Reihenfolge der Vergleichsprozesse Aspekt - Muster festgelegt wird.

4.2 Kontext der Muster

Die Lösung des für die Sensorimotorik spezifischen Problems, für die Verfolgung von Zielen geeignete, über Muster wahrnehmbare Ausgangsobjekte zu finden, wird dadurch erschwert, daß diese häufig gar nicht offenkundig, sondern "hinter" anderen Objekten verborgen sind. Darum sind Indizien erwünscht, die dem Suchprozeß Anhaltspunkte bieten. Dazu wird postuliert, daß die Objekte im allgemeinen in einer für sie charakteristischen Umgebung vorkommen, d.h. mit großer Wahrscheinlichkeit zusammen mit bestimmten anderen Objekten auftreten. Die entsprechenden Muster erscheinen in einem charakteristischen Kontext. Dieser soll in die heuristische Strategie einbezogen werden.

Im Prinzip sollen dazu die F_l, F_m, F_n usw. Kontextmuster gespeichert enthalten, die bei dem Vergleichsprozeß mit den Umweltmustern ebenfalls abgefragt werden. Wird beispielsweise das zu F_l gehörige $(e_j)_l$ nicht gefunden, wohl aber die bei F_l gespeicherten Kontextmuster, so ist dies ein Indiz dafür, daß die gesuchte Operation über F_l führt und deshalb ein Fortschreiten zu den vorgeordneten F_c und F_d eine hohe Erfolgswahrscheinlichkeit besitzt.

Als Kontext werden also diejenigen Muster bezeichnet, die von den in Rede stehenden Operationen nicht betroffen, also von ihnen nicht verändert werden, wohl aber gemeinsam mit den betroffenen Mustern auftreten.

Es lassen sich zwei Arten von Kontext unterscheiden:

- der allgemeine Kontext
- der bestimmte Kontext.

Der <u>allgemeine</u> Kontext ist die allgemein zusammen mit einem Aspekt F_i wahrgenommene Kontextinformation. Sie ist für dessen Identität in gewissem Sinne ebenso charakteristisch wie seine durch ${}_AM_{ir}$, F_r bezeichneten Transformationseigenschaften. Sie werden deshalb ebenfalls als Pointer in F_i eingetragen. Zusammengefaßt lauten die Eintragungen bis jetzt also

$$F_i \left[{}_AM_{ir},\ F_r,\ w(r),\ H(r),\ \overline{F_k'},\ F_K\right]$$

${}_AM_{ir}$	Operator und Pointer zur Aktion A
F_r	Zielaspekt
$w(r)$	Maß für die Zuverlässigkeit oder "Eingeübtheit" der Transformation
$H(r)$	Maß für die Häufigkeit der Ausführung der Transformation als Hilfsgröße für die progressive deepening strategy

$\overline{F_k'}$ Negation

F_K allgemeiner Kontext als indikatorische Hilfsgröße bei Suchprozessen.

Die Wichtigkeit von F_K für die Identität von F_i kann dazu führen, daß bei Auftreten eines Kontextes F_L statt F_K Differenzierungsprozesse analog denen der Abb. 2 eingeleitet werden können, die durch die Nichtübereinstimmung des durch die Transformation erzeugten Musters mit $(e_j)_r$ ausgelöst werden.

Die allgemeine Kontextbeziehung ist reversibel: Ist F_K Kontext für F_i, so ist umgekehrt F_i Kontext für F_K. Sie charakterisiert die Verknüpfung von zwei Aspekten. Ein Spezialfall einer solchen Verknüpfung kann der durch Zusammensetzung von F_i und F_K gebildete Aspekt $F_i \cup F_K$ mit höherem p sein, der als wesentliches zusätzliches Kennzeichen bestimmte eindeutige Transformationseigenschaften besitzt. Hier definiert $F_i \cup F_K$ ein besonderes gegenseitiges Kontextverhältnis.

Die Funktion der allgemeinen Kontextverknüpfung für den Suchprozeß kann auch umgekehrt werden. So wie das Zur-Deckung-bringen des Kontextaspektes mit einem empfangenen Muster auf ein gesuchtes Muster hinweist, kann auch ein empfangenes Muster über eine Kontextverknüpfung einen Aspekt aufrufen und damit nach einem Vorwärtsschema eine Reaktion auslösen. Damit wird aus der Kontextverknüpfung ein <u>bedingter Reflex.</u> Und in der Tat können in Pawlows klassischem Beispiel Glockenzeichen und Anblick des Futters als gegenseitiger Kontext verstanden werden. Die biologische Funktion des Kontextes ist eine Erweiterung der Bandbreite der Wahrnehmung, sei es, um die erforderlichen Reaktionen zuverlässiger auszulösen, sei es, um Hilfen im Suchvorgang zu haben.

Der <u>bestimmte</u> Kontext ist dagegen ein Aspekt, dessen Verknüpfung mit einem im laufenden Prozeß bearbeiteten oder gesuchten Aspekt das System

irgendwann in der Vergangenheit explizit wahrgenommen hat. Das System weiß aus den vorangegangenen Prozessen um den Zusammenhang zwischen den zugehörigen Objekten und kann beliebig in der Zukunft davon Gebrauch machen. Ist z.B. festgestellt worden, daß ein bestimmtes Objekt hinter, unter, in einem anderen versteckt ist, so wird das System genau dort suchen, wenn es das Objekt wieder benötigt. Wie der allgemeine erlaubt der bestimmte Kontext eine Wahrnehmung unter Zuhilfenahme einer anderen, offenkundigeren. Bezeichnet der allgemeine Kontext eine Verknüpfung mit einer gewissen, unter 1 liegenden Wahrscheinlichkeit, so wird die Verknüpfung im bestimmten Kontext als gegeben unterstellt, auch wenn sie in Kategorien des allgemeinen Kontextes unwahrscheinlich ist. Weist ein bestimmter Kontext auf das Vorhandensein eines Objektes hin, so kann das System die daraus folgenden Handlungen auch unternehmen, ohne vorher mit den Sinnesorganen die tatsächliche Existenz des Objektes überprüft zu haben. Ein Mensch, der am Morgen einen Geldschein eingesteckt hat, bestellt sich auch seelenruhig ein Mittagessen, ohne nochmals nachgesehen zu haben, ob das Geld auch da ist.

Es ist nicht zweckmäßig, den bestimmten Kontext ebenfalls in Verbindung mit den jeweiligen Aspekten zu speichern, da er möglicherweise unwahrscheinlich und darum nicht unbedingt charakteristisch für diese ist. Es empfiehlt sich vielmehr eine lineare Aufzeichnung nach der zeitlichen Reihenfolge, die das folgende Aussehen hat:

$$\begin{array}{ccccccccc} a & b & c & d & e & f & g & & \\ F_1 & F_2 & {}_AM_{23} & F_3 & {}_BM_{34} & F_4 & F_5 & F_6 & \ldots \end{array}$$

a, b, c, d usw. sind die Adressen, unter denen die wahrgenommenen Aspekte in diesem Speicher stehen, die F_1, F_2, ... und die ${}_AM_{12}$ usw. sind Pointer an die entsprechenden Plätze im F-Speicher bzw. die Aktuatoren. Es wird einfach eine Folge von Aspekten, ihre Gleichzeitigkeit und ihre Verknüpfung durch verschiedene Operationen zu Protokoll genommen und dieser Speicher soll deshalb als <u>Protokollspeicher</u> bezeichnet werden.

Ein Stück aus dieser fortlaufenden Kette von Pointern bezeichnet einen bestimmten Sachverhalt. Um im Zuge der heuristischen Prozesse von diesem Sachverhalt Gebrauch machen zu können, wird seine Adresse ebenfalls bei dem betreffenden Aspekt gespeichert, z.B. d bei F_3 und die vollständige Eintragung lautet jetzt

$$F_3 \left[{}_A M_{3r}, F_r, w(r), H(r), \overline{F_k'}, F_K, d \right]$$

Ändert sich der durch d bezeichnete Sachverhalt, so muß die Eintragung geändert werden. Die biologische Bedeutung des Protokollspeichers liegt auf der Hand. Erst durch ihn wird ein zielgerichtetes Handeln auch über längere Zeiträume hinweg möglich auch dann, wenn die Objekte des Handelns nicht immer sichtbar sind. Nur so kann der Ablauf einer heuristisch verfolgten längeren Kette aus Aspekten und Operationen hinreichend überwacht werden. Auch bei den Prozessen auf der sensorimotorischen Ebene kommt ihm bereits eine besondere Bedeutung zu. Becker [6] benutzt kein derartiges System. Das "short term memory", das dort auch eine lineare Folge von Schemata aufnimmt, hat eine völlig andere Funktion.

4.3 Einfache und zusammengesetzte Aktionen

Die Aktionen sind bisher nur sehr pauschal behandelt worden. Das Werkzeug, z.B. der Manipulator, mit dem das System in seine Umwelt eingreift, hat im allgemeinen mehrere Freiheitsgrade der Bewegung, für jeden sei ein elektrischer Antriebsmotor vorhanden, der ein- und ausgeschaltet werden kann und möglicherweise noch verschiedene Drehzahlstufen besitzt. Die Schalter, die die Motoren A, B, C usw. einschalten, seien über Adressen A_k, B_k usw. erreichbar und zu betätigen. Die k seien fortlaufende Indizes und erlauben die Schaltung auf verschiedene Geschwindigkeiten. Beispiel: Es gebe drei Schalteradressen A_1, A_2, A_3.

Es bedeute:

$A_1 = 1$
$A_2 = O$
$A_3 = O$ der Motor A läuft mit Drehzahlstufe 1

$A_1 = 1,\ A_2 = 1,\ A_3 = O$ der Motor A läuft mit Drehzahlstufe 2

$A_1 = 1,\ A_2 = 1,\ A_3 = 1$ der Motor A läuft mit Drehzahlstufe 3

$A_1 = A_2 = A_3 = O$ der Motor A steht.

Die jeweilige Gestalt des Musters A_1, A_2, A_3 (dessen Kodierung selbstverständlich auch anders gewählt werden kann) bezeichnet dann den Ablauf der Aktion A. Entsprechendes gebe es für die anderen Motoren B, C usw.

Der Prozeß der Abb. 1, bei dem durch eine Aktion Aspekte formiert werden, hat dann in größerem Detail den folgenden Ablauf:

- Start einer Aktion, z.B. $A_1 = 1$, $A_2 = A_3 = O$ oder kurz $A = 1OO$

- Auftreten der Komponenten e_m, e_n in einem für die Veränderung charakteristischen Vektor $(e_j)_i$
  kurz $(e_j)_i = 1$

- Bildung von $(e_j)_i$ und Anwahl von F_i

- Ende der Aktion,
  kurz $A = OOO$

- Bildung von $(e_j)_r$ und Anwahl von F_r

- Zwischenprozeß zur ersten Generalisation von F_i und F_r:
  Bildung $M_{ir} = F_r - F_i$
  Bildung $F_i^* = F_i \cap M_{ir}$
  $F_r^* = F_r \cap M_{ir}$

Die einfacheren F_i^*, F_r^* übernehmen die Rolle der F_i, F_r

- Eintragung der Pointer.

  In F_i: F_r, Anfangswert von w(r)
  ${}_AM_{ir}$ als Adresse in einem <u>motorischen Speicher</u>.
  Die übrigen Eintragungen A(r), $\overline{F_k'}$, F_K, d sind in dieser Anfangsphase noch nicht zu definieren.

  In F^r: F_i (Rückwärtsschema)

  In ${}_AM_{ir}$: A = 1OO, $(\dot{e}_j)_i$, A = OOO

Die weiteren Einübungsprozesse ergeben sich gemäß Abb. 2. Es ist möglich, aus beobachtetem und in ${}_AM_{ir}$ eingetragenem $(\dot{e}_j)_i$ eine Korrekturgröße herzuleiten, die unmittelbar auf die Drehzahl von A zurückwirkt und so aus einer Ein-Aus-Bewegung zu einer geregelten Bewegung zu kommen. Durch den Index von $(\dot{e}_j)_i$ ist in bekannter Weise der Grad an Differenziertheit auszudrücken, mit dem dies Muster reproduziert werden muß.

Die so zu spezifizierenden Aktionen sind sehr einfacher Natur wie das Heben des Manipulatorarms, das Öffnen oder Schließen der Hand usw. und die Registrierung dieser Bewegungen und der sie begrenzenden Zustände durch die Sensoren. In der ersten Phase legt sich das System einen Vorrat an derartigen Aktionen zu.

Mittels der heuristischen Zielverfolgung gemäß Abb. 5 können aus den einfachen Aktionen kompliziertere zusammengesetzt werden. Voraussetzung dafür ist die Setzung eines Ziels F_r. In der Natur ist die Erklärung eines Aspektes zum Ziel häufig a priori durch eine biologische Funktion eingebaut; im künstlichen System kann dies (und nur dies) durch menschlichen Befehlseingriff erfolgen.

Ist als Folge des Suchprozesses schließlich die Transformation $F_g \rightarrow F_n \rightarrow F_r$ erfolgreich durchgeführt worden, so kann eine neue Eintragung

$F_g \left[M_{gr}, F_r, \ldots \right]$ Vorwärtsschema

$F_r \left[F_g, \ldots \right]$ Rückwärtsschema

$M_{gr} \left[{}_A M_{gn} \, F_{nB} \, M_{nr} \right]$

gemacht werden. Der Operator M_{gr} setzt sich aus zwei, unmittelbar die Motoren A und dann B betätigenden und zuvor beschriebenen Elementaroperatoren ${}_A M_{gn}$ und ${}_B M_{nr}$ zusammen.

Eine Fortsetzung ist beliebig möglich, aus M_{gr} und einem M_{ag} kann ein $M_{ar} \left[M_{ag} \, F_g \, M_{gr} \right]$ und so schließlich eine ganze Hierarchie von Operatoren aufgebaut werden. Charakteristisch ist, daß im motorischen Speicher, der die M_{xy} usw. aufnimmt, an jeder Adresse zwei Operatoren einen Aspekt einrahmen. Auf der untersten Stufe stehen die direkt einen ganz bestimmten Motor betätigenden Operatoren. An diesen Adressen wird ein Bewegungsaspekt von zwei Schaltbefehlen eingerahmt und eine Rückkopplung zwischen beiden ist möglich.

Der Ablauf bei Aufruf von z.B. $M_{ar} \left[M_{ag} \, F_g \, M_{gr} \right]$ ist der folgende:

- $M_{ag} \left[M_{ab} \, F_b \, M_{bg} \right]$ wird angewählt
- $M_{ab} \left[{}_A M_{aq} \, F_q \, {}_B M_{qb} \right]$ wird angewählt
- ${}_A M_{aq} \left[A = 1 \;\; F_a \;\; A = O \right]$ wird ausgeführt
- F_q wird mit ankommendem $(e_j)_q$ zur Deckung gebracht
- ${}_B M_{qb} \left[B = 1 \;\; \dot{F}_q \;\; B = O \right]$ wird ausgeführt

- F_b wird mit ankommendem $(e_j)_b$ zur Deckung gebracht.
  Damit ist M_{ab} ausgeführt.

- M_{bg} wird ebenso bis herunter zu den Elementaraktionen ausgeführt, die Aspekte werden mit den empfangenen Mustern zur Deckung gebracht.
  Damit ist M_{ag} ausgeführt.

- M_{gr} wird ausgeführt.

M_{ar} entspricht also der Kette.

$$M_{aq}\ F_q\ M_{qb}\ F_b\ M_{bg}\ F_g\ M_{gr}$$

und die einzelnen Operatoren enthalten ihrerseits noch die jeweiligen Bewegungsaspekte. Abb. 6 zeigt das als hierarchische Struktur. Die gesamte zusammengesetzte Aktion ist also stark an die jeweiligen sensorischen Signale rückgekoppelt.

Im Zuge der weitergehenden Generalisierung und Einübung können manche der Zwischenaspekte auf sehr einfache Grundformen mit kleinem Index abgebaut werden. Grenzfall: $p \longrightarrow O$, die Wahrnehmung des betr. Zwischenaspektes ist für den Ablauf der zusammengesetzten Operation nicht erforderlich.

Der in Abschn. 4.2 beschriebene Protokollspeicher enthält ähnliche MFM-Ketten, wie sie hier auftreten. Es liegt nahe, einen Umordnungsprozeß zu definieren, der eine Folge $M_{12}\ F_2\ M_{23}$ dann durch das M_{13} ersetzt, wenn das System ein solches definiert hat. Unter Zuhilfenahme des motorischen Speichers ist es möglich, die Details von M_{13} zu rekonstruieren. Gegebenenfalls bei F_2 stehende Kontexte sind dann allerdings endgültig "vergessen", soweit sie nicht zum bei F_2 abgespeicherten allgemeinen Kontext gehören.

4.4 Adressenspeicher und assoziative Speicher

Bei den üblichen Speichern kann die enthaltene Information dann ausgelesen werden, wenn die betreffende Position über die bekannte Adresse angewählt wird. Das beschriebene System erfüllt diese Voraussetzungen dadurch, daß das empfangene Muster als Code für die Adresse im Aspektspeicher benutzt wird und von dort weiterführende Pointer die Adressen in den weiteren Speichern angeben. Die Rückwärtsschemata erlauben einen Speicherzugriff auch nach anderen "key-words". Im Prinzip ist das möglich, in der Hardwareausführung ergeben sich Nachteile.
Der größte Nachteil einen solchen Systems wäre, daß im Aspektspeicher für n Eingangskanäle 2^n Speicherplätze vorzusehen wären, in denen Raum für ziemlich viele Informationen sein müßte. Nur ein Teil der Plätze würde wirklich benutzt werden. Bereits bei sehr geringem n würde man an die technisch realisierbare Grenze stoßen selbst dann, wenn nicht Kernspeicher benutzt würden.

Hier bietet ein Zuordnungsspeicher Abhilfe, der die ankommenden $(e_j)_i$ mit den physikalischen Adressen Q_i der Aspekte F_i verbindet. Solch ein "Inhaltsverzeichnis" besteht im Prinzip aus zwei Spalten, in der ersten steht $(e_j)_i$, in der zweiten Q_i. Die $(e_j)_i$ seien nach dem Index in Gruppen eingeteilt, innerhalb der Gruppen kann die Reihenfolge durch die Dualzahl gegeben sein, als die man $(e_j)_i$ lesen kann.

Beispiel für $n = 4$:

$(e_j)_i$	Q_i		
0001	+	)	
0010	+	)	
0100	+	)	$p = 1$
1000	-	)	
0011	+	)	
0101	-	)	
0110	+	)	$p = 2$
1001	-	)	
1010	-	)	
1100	-	)	

$(e_j)_i$	Q_i		
O111	+	)	
1O11	-	)	p = 3
11O1	-	)	
111O	-	)	
1111	-		p = 4

Alle mit + bezeichneten Q entsprechen einem relevanten F_i . Der Speicher kann kondensiert werden auf

OOO1	Q_i
OO1O	Q_j
O1OO	Q_k
OO11	Q_l
O11O	Q_m
O111	Q_n

Steht O1OO an, so kann das $Q_k \longrightarrow F_k$ durch einen Suchprozeß in der Gruppe p = 1 rasch gefunden werden. Wird ein neuer Aspekt, z.B. entsprechend O1O1 gebildet, so muß der ganze Speicher neu geschrieben werden, um zwischen Q_l und Q_m die entsprechende Eintragung zu machen. Solche Operationen sind möglich; für den sofortigen Bedarf kann außerdem zwischen den einzelnen Zeilen ein gewisser Zwischenraum gelassen werden. Da das gleiche F_i mehreren Transformationen unterworfen sein kann (s. Abb. 3), kann es mehrere Q_i geben, die auf je ein F_i mit einem anderen Inhalt hinweisen.

Eine elegantere Methode bieten die assoziativen Speicher, die hardwaremäßig nicht so weit entwickelt sind. Sie haben eine Matrixstruktur; die Zeilen der Matrix sind die miteinander verknüpften Wortgruppen. Wird eine bestimmte Wortgruppe oder auch ein einzelnes Wort gesucht, so wird es in ein Abfrageregister eingelesen. Die Zeile, in der das gesuchte Wort steht, gibt dann ein Signal. Abb. 7 zeigt die Matrix für den Aspektspeicher. Wird $(e_j)_i$ in das Abfrageregister eingelesen, so kann sofort, falls $(e_j)_i$ im Speicher steht, die gesamte damit verknüpfte Information

M_{ir}, F_r, F_K, w(r), H(r), $\overline{F_k'}$ ausgegeben werden und z.B. der Befehl M_{ir} ausgeführt werden. Da im allgemeinen mehrere Befehle mit $(e_j)_i$ assoziiert sind, können die Auswahlkriterien durch weitere Eingabe von $(e_j)_r$, $\overline{F_k'}$, F_K, w(r) usw. beliebig verschärft und so eindeutige Operationen erzielt werden.

Die Wortgruppen können in beliebiger Reihenfolge im Speicher stehen, da diese nicht mehr entscheidend ist. Die Unterscheidung in Vorwärts- und Rückwärtsschemata und ihre getrennte Speicherung wird gegenstandslos. Die Speicherung der Adresse d des Protokollspeichers wird auch nicht mehr benötigt, wenn dieser ebenfalls assoziativ aufgebaut wird und die Prozesse mit einer fortlaufenden Zeitskala verknüpft. Der bestimmte Kontext läßt sich unmittelbar abfragen, ein Pointer ist überflüssig.

Es gibt Konstruktionsvorschläge für assoziative Speicher auf kryotechnischer, magnetischer, Halbleiter-, ferroelektrischer und elektronenoptischer Basis [16]. Nach dem Stand der Technik scheint es jedoch noch nicht möglich zu sein, das gesamte Speichervolumen in dieser Form zu erträglichen Kosten zur Verfügung zu stellen. Ein interessanter Kompromiß wäre es aber, wenn für die heuristischen Prozesse nach Abb. 5 die jeweils mit dem Zielaspekt über Operatoren verbundenen übrigen Aspekte aus dem großen Adressenspeicher in einen kleinen assoziativen Speicher eingelesen werden könnten. Die beschriebenen Prozesse, bei denen nacheinander viele Aspekte auf Übereinstimmung mit einem einfallenden Muster abgefragt werden müssen, können so durch einen Simultanvorgang ersetzt werden. Dadurch ist es rascher möglich, ein gesuchtes Objekt in der Umwelt zu finden.
Auch eine softwaremäßige Darstellung der Assoziativspeicher durch die HASH-Technik [21] ist denkbar.

4.5 Zusammengesetzte Aspekte oder Konzepte

Die Einführung der zusammengesetzten Aktion und der entsprechenden Operatoren erlaubt auch eine Verallgemeinerung des Aspektbegriffes.

Stellt eine Kette

$$M_{ab}\ F_b\ M_{bc}\ F_c\ M_{cd}$$

einen zusammengesetzten Operator, dem eine zusammengesetzte Aktion zugeordnet ist, dar, so kann die Kette

$$F_a\ M_{ab}\ F_b\ M_{bc}\ F_c\ M_{cd}\ F_d$$

als ein <u>zusammengesetzter Aspekt</u> aufgefaßt werden. Und so, wie der oben angegebene Operator im M_{ad} abgekürzt wird, so soll der zusammengesetzte Aspekt

$$F_a\ M_{ad}\ F_d \quad \text{als} \quad F_{ad}$$

bezeichnet werden.
Zusammengesetzte Aspekte sollen <u>Konzepte</u> heißen.

Die Eintragungen in den einzelnen, das Konzept ausmachenden Aspekten haben im wesentlichen die bereits beschriebene allgemeine Struktur, z.B.

$$F_b\ \left[M_{bc},\ (e_j)_c,\ w(c),\ H(c),\ \overline{F_k'},\ \ldots\right]$$

Die drei Punkte bezeichnen die Stelle, an der Kontext und Protokolladresse zu stehen hätten. Da diese im Sinne der Generalisierung zum gesamten Konzept und nicht zu den Teilaspekten gehören, entfallen sie hier.
Statt dessen ist eine logische Bedingung notwendig, die die Ausführung der Operation M_{bc} außer wie üblich von dem Vorhandensein von $(e_j)_b$ und dem Nichtvorhandensein von $(e_j)_k'$ noch zusätzlich von der vorherigen Ausführung der Operation M_{ab} abhängig macht. Diese Vorschrift, die als und-Bedingung eine ähnliche Funktion wie die Negation $\overline{F_k'}$ hat, wird nach dieser eingetragen:

$$F_b\ \left[M_{bc},\ F_c,\ w(c),\ H(c),\ \overline{F_k'},\ M_{ab}\right]$$

Der letzte Aspekt des Konzeptes, hier z.B. F_d, müßte dann die Reaktion des Systems als Antwort auf die Erkennung des gesamten Konzeptes enthalten. Statt der sonst üblichen Antwort, z.B. M_{dx}, F_x, w(x), H(x) steht hier ein Pointer F_{ad} an den dem Konzept zugeordneten Speicherplatz (ggf. in einem besonderen Speicher, dem Konzeptspeicher).

Also: $F_d \left[- , F_{ad}, - , - , - , \overline{F_1}, M_{cd} \right]$

Die Leerpositionen deuten an, daß außer M_{dx} auch w(x) und H(x) hier nicht benötigt werden.

F_{ad} enthält als Eintragung dann die eigentliche Reaktion und neue Zieladresse, z.B.

$$F_{ad} \left[M_{dx}, (e_j)_x, w(x), H(x), - , F_K, F_a, F_d \right]$$

Hier sind w(x) und H(c) wieder sinnvoll, das Problem der Negation (Leerstelle) ist durch die enthaltenen primären Aspekte erledigt, der Kontext F_K hat eine Bedeutung und für die Rückadressierung sind das Anfangsglied F_a und das Endglied F_d der ursprünglichen Kette angegeben.

Jeder der in F_{ad} stehenden Pointer kann seinerseits wieder auf zusammengesetzte Aspekte und Operatoren weisen, umgekehrt kann ein neues Konzept F_{ay} aus F_{ad} M_{dx} F_{xy} aufgebaut werden, so daß eine Konzepthierarchie entsteht.

Der Begriff des zusammengesetzten Aspektes ist besonders dann interessant, wenn die in seinem Aufbau enthaltenen Aktionen nicht irgendeine Bewegung des Manipulators bedeuten, sondern eine Betätigung der Sensoren. Diese kann in einem Schwenk der Kamera bestehen, in einer Änderung der Fokussierung, einem sukzessiven Nachfahren der Konturen eines Objektes oder in einer Veränderung des Ausschnitts. Die Aktionen können in einer mechanischen Bewegung resultieren oder elektronisch internalisiert sein, ja sie können in der Ausführung mathematischer Operationen

wie der Bildung räumlicher und zeitlicher Differentialquotienten bestehen. Solche Aktionen werden als sensorische Aktionen bezeichnet.

Durch die Verbindung von Konzepten aus Aspekten und sensorischen Aktionen entstehen neue Wahrnehmungsinhalte von unbegrenzter Vielfalt. Die Aspekte als die Wahrnehmungen auf den "ersten Blick" lassen sich beliebig kombinieren oder detaillieren durch die Folge von sensorischen Aktionen, durch die ein Objekt in ein inneres Bild überführt wird, das alle für sein Verhalten wesentlichen Eigenschaften enthält.

Aiserman [17] unterscheidet die geometrische und die strukturelle Methode der Mustererkennung. Die geometrische Methode, die Abgrenzung von Bereichen im Hyperwürfel, wird hier durch die Aspektadressen und ihre Organisation erreicht. Die Aspekte aber sind ein Alphabet von einfachen Grundmustern, aus denen die Konzepte als Worte, die komplexeren Mustern entsprechen, zusammengesetzt werden. Das entspricht der strukturellen Methode, zu der heute erste Ansätze in Arbeit sind. Darüber hinaus werden auch die relevanten Verknüpfungen von Aspekten zu Konzepten durch selbsttätige Lernprozesse entwickelt.

Becker's [6] "secondary Kernels" sind den Konzepten recht ähnlich. Da er jedoch den Prozeß der Verarbeitung und Unterscheidung der Muster und die hier dazu entwickelte Muster- (= Aspekt) Hierarchie nicht behandelt, wird die große Bedeutung der Konzepte für den sublimeren Erkennungsprozeß nicht herausgearbeitet. Die sensorische Informationsaufnahme, ihre Klassifizierung und das Verhalten des Systems mit seiner Zielverfolgung sind auf allen Ebenen intim miteinander verflochten, so daß eine isolierte Behandlung von Teilproblemen wegen der gegenseitigen Rückwirkungen nicht zum Ziele führen kann.

Das wird besonders deutlich, wenn im nächsten Abschnitt die Auswirkung des Konzeptbegriffes auf die primäre Kodierung der Muster behandelt werden wird.

4.6 Das Invariantenproblem der Mustererkennung

Die Muster, die die Sensoren eines sich in einer realen Umwelt bewegenden Systems empfangen, sind vielfachen Variationen unterworfen, ohne daß sich dadurch ihre Bedeutung, d.h. die jeweils zweckmäßige Reaktion des Systems, ändert.

Zu den wichtigsten Variationen gehören:

- Translationen relativ zu den Sensoren, wie etwa die Verschiebung eines Bildes auf der Netzhaut, aber auch eines Klangmusters auf der Frequenzskala
- Affine Transformationen durch wechselnde Entfernung zwischen Objekt und Sensor
- Rotationen
- Deformationen, wie z.B. bei Schriftzeichen.

Über diese Transformationen muß das System generalisieren; und über die Generalisation muß ein invarianter Operator aufgerufen werden. Generalisation und Invariantenbildung sind verschiedene Seiten des gleichen Problems.

Rosenblatt [18] zählt vier Möglichkeiten der Generalisation auf:

1. die analytisch-deskriptive Methode
2. die Bildtransformation
3. die zusammenhängende Generalisation
4. die Transformations-Assoziation.

Die analytisch-deskriptive Methode (z.B. [19] filtert von vornherein analytisch zu beschreibende Invarianten aus dem Muster bzw. erzeugt sie durch geeignete Operationen - es entsteht gewissermaßen eine Tabelle der Mustereigenschaften, die gegenüber Transformation invariant ist.

Die Bildtransformation, die etwa in einer der ältesten Arbeiten zur Generalisation [20] zugrunde liegt, besteht im Prinzip aus einer fortwährenden Mittelung über die einzelnen Transformationen; die Mittelwerte sind dann die Invarianten. Rosenblatt zeigt die Schwächen dieses Prozesses in Form des Verlustes an relevanter Information und des großen Hardwareaufwandes.

Die zusammenhängende Generalisation entspricht etwa der einheitlichen Klassifizierung zusammenhängender Bereiche oder Cluster im Hyperwürfel.

Die Assoziation transformierter Muster erklärt sich selbst; einfache Grundformen studiert Rosenblatt an bestimmten Versionen seines Perceptrons.

Das hier entwickelte Strukturprinzip von einfachen und zusammengesetzten Aspekten, einfachen und zusammengesetzten Operatoren und ihre Ausbildung und Abgrenzung in der Wechselwirkung mit der Umwelt erlaubt einen Lösungsweg für das Invariantenproblem, der wesentliche Züge der Möglichkeiten 1, 3 und 4 kombiniert. Davon ist 3 in der Beschreibung der Aspekthierarchie und -gruppierung schon sehr weitgehend beschrieben worden. Dabei ist das Problem der sich gegenseitig durchdringenden Cluster ein wesentlicher Gesichtspunkt, der nach Kenntnis des Autors in der Literatur noch nicht behandelt worden ist.

Die Grundlage der Invariantenbildung ist die zweckmäßige Kodierung der sensorischen Signale. In Abschn. 2 waren dafür in ziemlich pauschaler Form Spezifikationen angegeben worden. Wenn es gelingt, in dem dort angedeuteten Sinne invariate Komponenten des Musters wie etwa die gegenseitige Lage, Richtung und Krümmung der Konturen als Komponenten in den Vektor $(e_j)_i$ einzubeziehen, ist die Aufgabe gelöst. Die dafür erforderlichen Aktionen wie die Bildung erster und zum Teil höherer Differentialquotienten nach Zeit, Ort und Intensität brauchen keinesfalls nach Maßgabe eine Analyse der zu erwartenden Muster durch den

Menschen vorgesehen werden. Vielmehr können sie im Sinne des vorhergehenden Abschnittes als sensorische Aktionen behandelt werden. Ein bestimmter Ablauf sensorischer Aktionen erzeugt in der beschriebenen Weise ein Konzept, in dem ein Pointer steht, und dieser übernimmt die Rolle des aus dem Muster zu bildenden Pointers $(e_j)_i$, der den ersten Aspekt aufruft. Im verallgemeinerten Sinne findet die Konzeptbildung also schon auf der ersten Stufe der Kodierung statt; auch hier können die wirklich zweckmäßigen Prozeßabläufe durch Lernvorgänge aus einem ungeordneten Vorrat an möglichen Aktionen synthetisiert werden.

Man kann zwei Aktionsschemata zur Erzeugung von Pointern $(e_j)_i$ mit möglichst vielen invarianten (und damit letztlich auch relevanten) Komponenten unterscheiden

- Syntheseaktionen
- Annullierungsaktionen.

Syntheseaktionen erzeugen einen zusammengesetzten Aspekt (hier interessiert nur der dadurch gewinnbare Pointer $(e_j)_i$ zum "eigentlichen" ersten Aspekt, der eine Antwort des Systems gespeichert hat). Das Schema FMFMF usw. entspricht z.B. der Bestimmung der Neiung einer Kontur, der Krümmung einer Kontur, der Zahl der an einem Schnittpunkt zusammenlaufenden Konturen usw. Von dort aus entwickeln sich kontinuierlich kompliziertere Konzepte, wie sie etwa durch das Nachfahren der Begrenzung eines Objektes mit den Sensoren entstehen. Der bedeutsame Punkt ist der, daß bei einem solchen System Aspekte und Konzepte etwas sind, was nicht etwa linear aufeinander und auseinander aufgebaut ist, sondern auf allen Ebenen von der Musteraufbereitung bis zu komplexen Aktionsabläufen einander durchdringt.

Nach einem ähnlichen Schema können aber auch Invarianten dadurch gebildet werden, daß im internen Prozeß eine Aktion, die zu einer

Transformation des Musters führt, von einer anderen überlagert wird, die jene genau annulliert. Ist ${}_{A}M_{ir}$ der Operator der Operation A allein, ${}_{B}M_{ri}$ der sie annullierenden Aktion B, so gilt offensichtlich

$$ {}_{A}M_{ir} \cup {}_{B}M_{ri} = O $$

Beispiele sind die Fixierung des Bildes eines bewegten Objektes auf der Netzhaut durch eine entsprechende Augenbewegung oder die Kompensation der Größenänderung eines Bildes je nach der Entfernung des Objektes.

Beide Möglichkeiten kombinieren im Grunde die analytisch-deskriptive Methode 1 der Invariantenerzeugung mit der Transformations-Assoziation nach 4. Analytisch-deskriptiv ist die Gewinnung invarianter Daten oder Komponenten; erzielt werden sie durch eine assoziative Aneinanderreihung von Aktionen bzw. Operationen nach Art der MFMFM-Ketten.

Die Lernfähigkeit bereits auf der Stufe der Kodierung scheint beim Menschen jedenfalls gegeben zu sein. Ein interessantes Beispiel ist die Entwicklung eines Gerätes zur Orientierung von Blinden, bei dem ein mit einer Fernsehkamera aufgenommenes Bild der Umgebung in ein auf den Rücken der Versuchsperson gegebenes Druckraster übersetzt wird. Nach einigem Training konnten die Versuchspersonen Einzelheiten der Umgebung "sehen" und auch Transformationen kompensieren. Zumindest dann, wenn die nachgeordnete Verarbeitungsstruktur gegeben war, konnte die Aufbereitung der taktilen Muster so weit gelernt werden, daß die Erkennungsprozesse funktionierten.

Eine in diesem Zusammenhang zu sehende Besonderheit der Invariantenerzeugung ist die für dies System zugrunde gelegte Bevorzugung der 1 vor der O in den Mustern, die die Aspektstruktur ermöglicht. Im Grunde sind es die gleichen Kodierungsprozesse, die neben der Erzeugung invarianter auch die Erzeugung relevanter Komponenten besorgen, so daß das System etwas damit anfangen kann. Wenn aber wichtige Information

nicht als 1, sondern als O erscheint und der Aspekt mit geringerem Index p spezieller ist als der mit großem p, funktionieren die beschriebenen Generalisierungs- und Differenzierungsprozesse nicht und das System kann in der Tat mit den Mustern nichts anfangen. Dann muß es genau so wie wenn die Muster durch Bewegungsvorgänge und dergleichen unkenntlich gemacht worden sind, durch Änderung der Kodierungsprozesse eine geeignetere Verschlüsselung gefunden werden. Der Aufbau von FMF-Ketten durch Lernprozesse bietet hierzu eine Möglichkeit.

Zusammenstellung einiger wichtiger Begriffe

(Seitenzahl in Klammern)

Aktion (12, 38)	Eingriff des Systems in die Außenwelt, der die von dort empfangenen Muster verändert.
Aspekt (14)	Einem bestimmten kodierten Muster zugeordneter Speicherplatz.
Oberaspekt F_o (zu F_k) (21)	Aspekt F_o, für den gilt $F_o = F_k \frown F_o$
Unteraspekt F_j (zu F_k) (21)	Aspekt F_j, für den gilt $F_j = F_k \smile F_i$ (F_i beliebig)
Aufruf, Anwahl (14)	eines Aspektes; erfolgt, wenn die Sensoren das der Adresse des Aspektes zugeordnete kodierte Muster aufrufen.
Differenzierung (15, 26 ff)	Hinzufügung weiterer relevanter Komponenten zu einem kodierten Muster - entspricht Erhöhung des Index eines Aspektes.
Generalisierung (15, 26 ff)	Fortlassung irrelevanter Komponenten aus einem kodierten Muster - entspricht Erniedrigung des Index eines Aspektes.
Index (14)	Zahl der logischen Einsen in dem die Adresse eines Aspektes angebenden kodierten Muster.
Kontext, allgemeiner (35)	Aspekte, mit denen gemeinsam ein bestimmter Aspekt häufig vorkommt.
Kontext, bestimmter (36)	Aspekte, mit denen gemeinsam ein bestimmter Aspekt in einem konkreten Fall vorgekommen ist und mit dem er im Protokollspeicher verzeichnet ist.

Konzept (46)	Zusammenfügung von Aspekten F und Operatoren M nach FMFMF usw.
Muster, kodiertes (9)	n-stelliger Binärvektor, der im Kodierer aus dem sensorischen Muster erzeugt wird.
Negation (22)	Oberaspekt eines aufgerufenen Aspektes, der nicht mit aufgerufen werden darf.
Operator (15)	Befehl, der von einem Aspekt aus den nächsten aufruft.
Protokollspeicher (37)	Aufzeichnung der ablaufenden Aspekt-Operator- bzw. Konzept-Operator-Ketten in linearer Anordnung entsprechend der zeitlichen Folge.
Schema (14)	An einer Aspektadresse eingetragene Folgeaspekte, Operatoren, Negationen, Kontexte, Hilfsfunktionen, die die Umwandlung des entsprechenden Musters durch eine Aktion in ein anderes beschreiben.
Vorwärtsschema	Schema, das an der Adresse des Ausgangsaspektes eingetragen ist.
Rückwärtsschema	Schema, das an der Adresse des Zielaspektes eingetragen ist.

Funktionen in den Schemata:

w(r) (13)	Erfolgsfunktion für Erreichen des Aspektes F_r. Wird bei jeder erfolgreichen Schemaausführung erhöht bis zum Grenzwert.
H(r) (35)	Maß für die Häufigkeit der Ausführung einer Operation, die auf F_r führt.

Literatur:

[1] J. Piaget, The Construction of Reality in the Child, Basic Books Inc., New York, 1954

[2] J. Piaget, Play, Dreams, and Imitation in Childhood, W. W. Norton & Co., Inc. New York 1951

[3] A. Bernstein et al. A Chess-Playing Program for the IBM 704, Proc. Western Joint Comput. Conf. 28, 157-159, (1958)

[4] A. I. Samuel, Some Studies in Machine Learning Using the Game of Checkers I, IBM J. Res. Developm. 3, 211-229, nachgedruckt in E. Feigenbaum, J. Feldman (eds) Computers and Thought, McGraw Hill, New York, II, IBM J. Res. Developm. 11, 6, 601-617

[5] G. W. Ernst, A. Newell, GPS, A Case Study in Generality and Problem Solving, Academic Press, New York, London 1969

[6] J. D. Becker, An Information-Processing Model of Intermediate-Level Cognition, Stanford Artificial Intelligence Project, Memo A1-119, AD 708082, May 1970

[7] A. G. Arkadev, E. M. Braverman, Teaching Computers to Recognize Patterns, Academic Press, London and New York, 1967

[8] S. Watanabe, ed., Methodologies of Pattern Recognition, Academic Press, New York, London, 1969

[9] F. Ratlift, Inhibitory Interaction and the Detection and Enhancement of Contours, Sensory Communication (W. A. Rosenblith ed.), p. 183-203, New York, MIT-Press and John Wiley and Sons, 1961

[10] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, W. H. Pitts, What the Frog's Eye Fells the Frog's Mind. (1959), Nachdruck in McCulloch, Embodiments of Mind, p. 230-255, MIT-Press, 1965

[11] F. Rosenblatt, The Perceptron, A Theory of Statistical Separability in Cognitive Systems Cornell Aeronautics Laboratory Inc., Rep. No. VG-1196-G-1, Office of Techn. Services PB 151247, (1958)

[12] R. G. Casey, G. Nagy, Advances in Pattern Recognition, Scientific American, 224, No. 4, 56-71, April 1971

[13] J. L. Phillips, Jr., The Origins of Intellect, Piaget's Theory, W. H. Freeman and Company, San Francisco, 1969

[14] S. Watanabe, Pattern Recognition as an Inductive Process, (1969), in [9], p. 521-534

[15] A. Newell, H. A. Simon, An Example of Human Chess Play in the Light of Chess Playing Programs, Carnegie Inst. of Tech., Pittsburgh, (1964)

[16] T. Feng, Large Scale Information Processing Systems, Vol. V, Study of Associative Memory Systems, AD 708729, National Technical Information Service, May 1970

[17] M. A. Aiserman, Remarks on Two Problems Connected with Pattern Recognition, in S. Watanabe (ed.) Methodologies of Pattern Recognition, Academic Press, New York, London (1969)

[18] F. Rosenblatt, Perceptual Generalisation over Transformation Groups, in Yorits, Cameron (ed.), Self-Organizing Systems, Pergamon Press, 1960

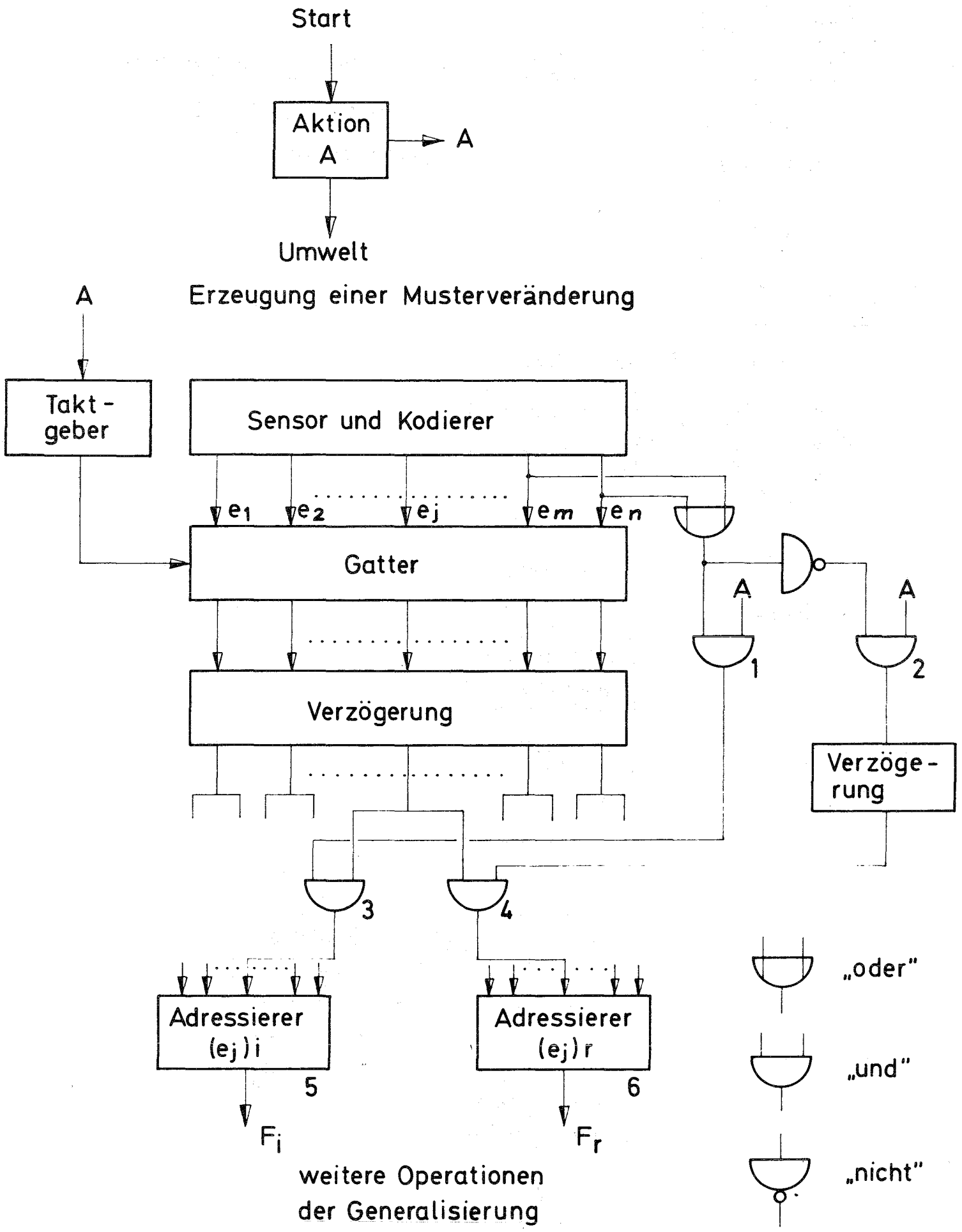
[19] K. Steinbuch, Automatische Zeichenerkennung, Nachrichtentechnische Zeitschr., 11, 210-219 und 237-244, (1958)

[20] W. Pitts, W. S. McCulloch, How we Know Universals, the Perception of Auditory and Visual Forms, Bull. Math. Biophysics 9, 127-147, (1947), Nachdruck in W. S. McCulloch, Embodiments of Minds, MIT-Press, 1965.

[21] R. Morris, Scatter Storage Techniques, Communications of the ACM, Vol. 11, No. 1, Jan. 1968, pp. 38-44

V. Y. Lum, P. S. T. Yuen, M. Dodd, Key-to-Address Transform Techniques: A Fundamental Performance Study on Large Existing Formatted Files.

Abb.1: Erzeugung relevanter Aspekte



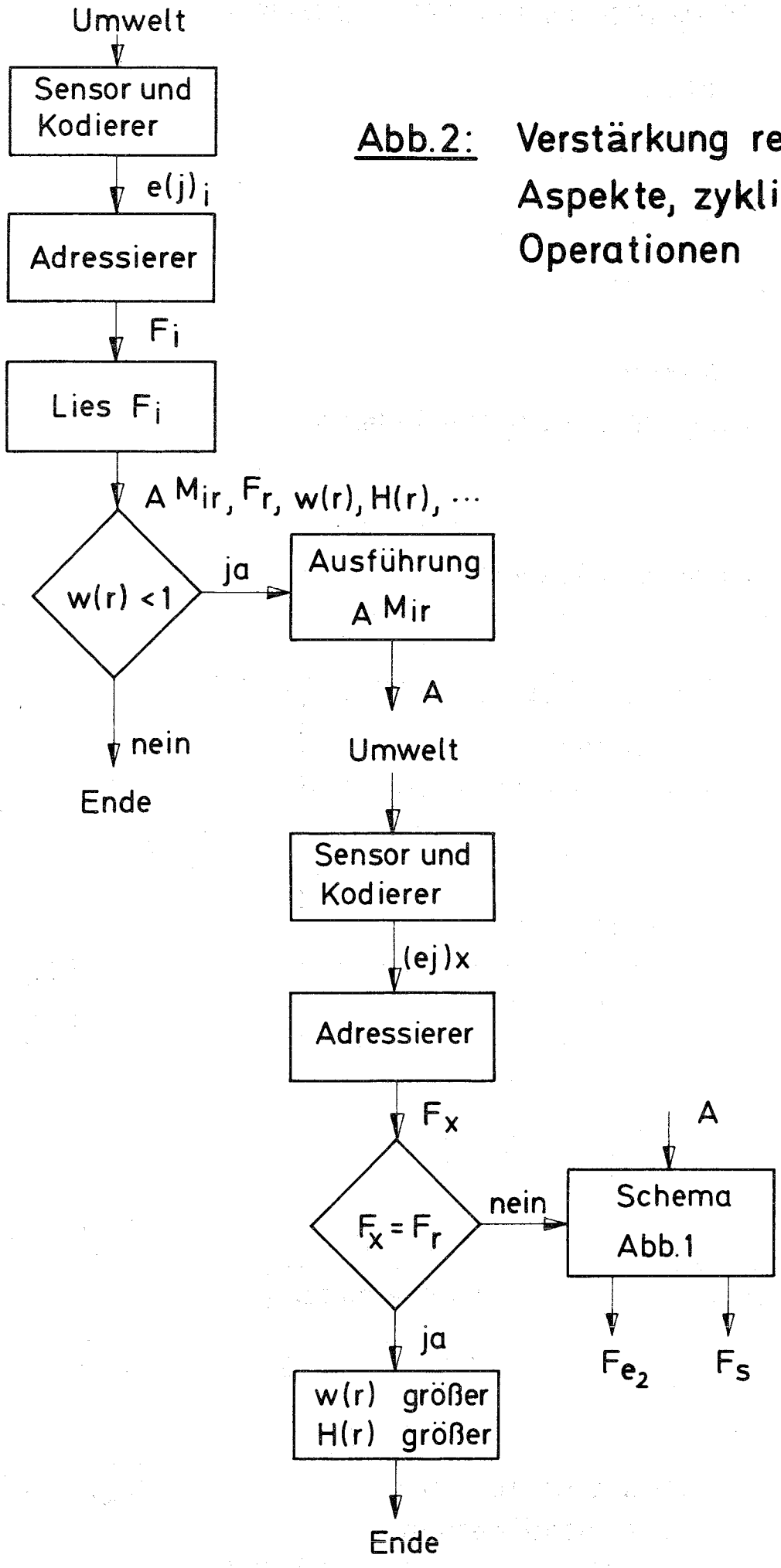
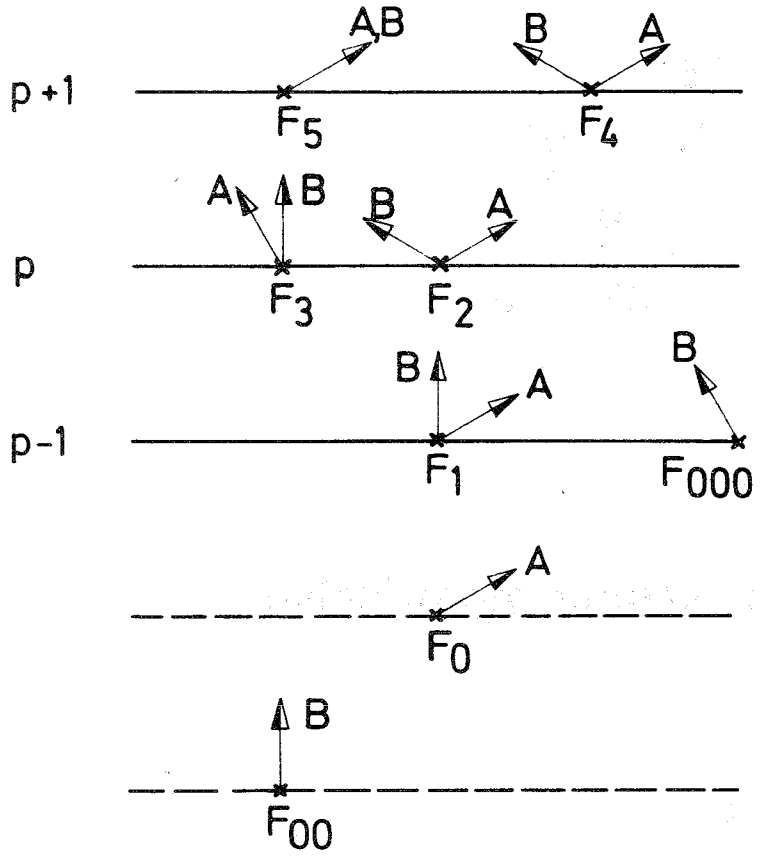


Abb.2: Verstärkung relevanter Aspekte, zyklische Operationen

Abb. 3: Aspekte mit Pointern



Die Richtung des Pfeiles veranschaulicht das Ergebnis der Transformation:

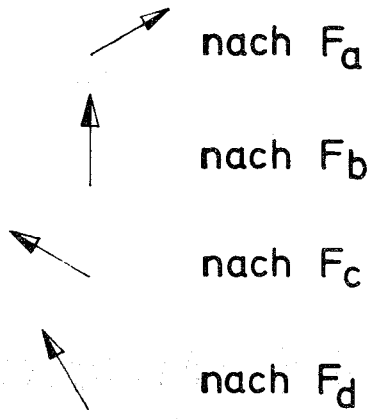


Abb. 4: Heuristischer Prozeß als Baumstruktur

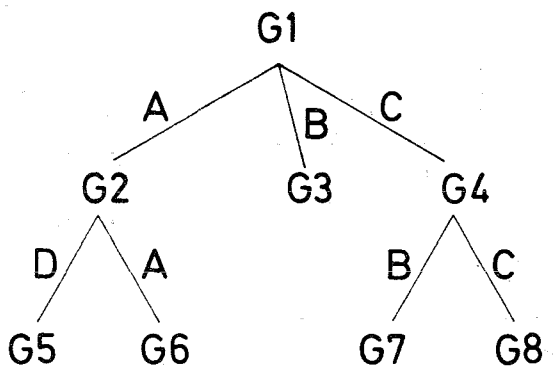


Abb. 5: Suchprozeß im sensorimotorischen Bereich

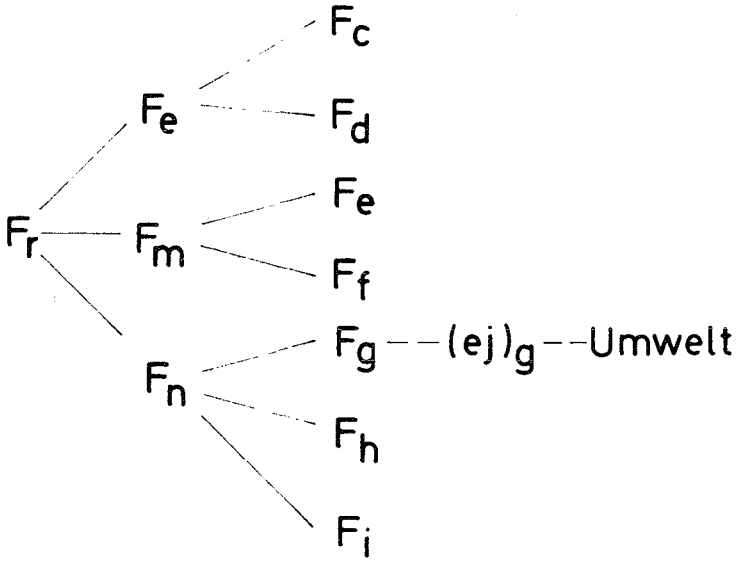


Abb.6: Struktur des zusammengesetzten Operators M_{ar}

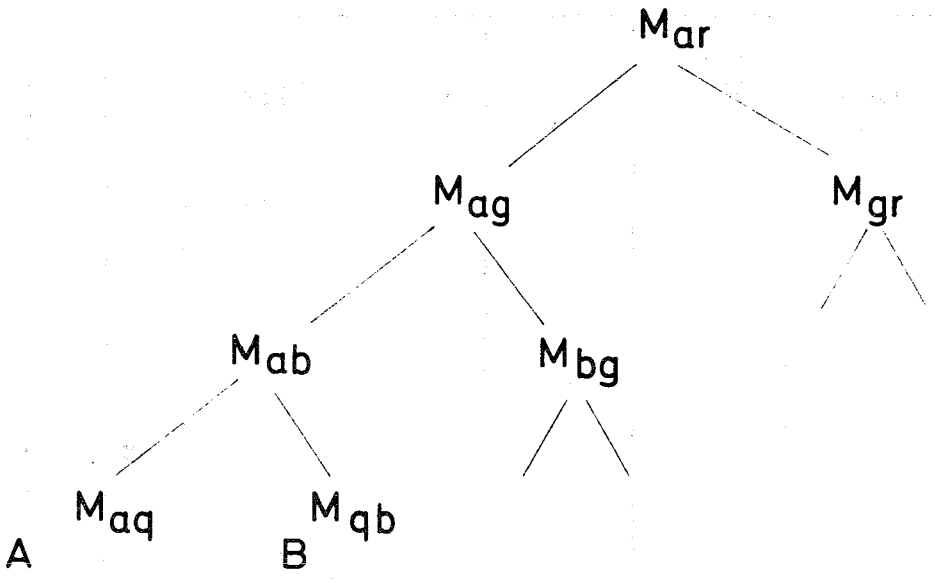


Abb.7: Anordnung der Aspekte in einem assoziativen Speicher