ANSGAR BECKERMANN

DENNETTS STELLUNG ZUM FUNKTIONALISMUS

1.

In einer Reihe von vielbeachteten Aufsätzen hat Daniel Dennett eine interessante Theorie über die Natur mentaler Zustände entwickelt, deren Grundbegriff der Begriff des *intentionalen Systems* ist. Dennett selbst hat dieser Theorie – etwas zögernd – in der Einleitung zu (1978a) den Namen "Intentionalismus" gegeben,[1] um sie auf diese Weise sowohl gegen die Identitätstheorie von Place und Smart als auch gegen den Funktionalismus von Fodor und Putnam abzugrenzen. Denn Dennett zufolge hat der Funktionalismus mit seiner These, mentale Zustände seien ihrer Natur nach funktionale bzw. logische Zustände bestimmter Systeme, ebenso unrecht wie die Identitätstheorie mit ihrer Auffassung, mentale Zustände seien mit bestimmten Gehirnprozessen identisch. Sein Hauptargument gegen den Funktionalismus lautet dabei:

> There is really no more reason to believe you and I 'have the same program' in *any* relaxed and abstract sense, considering the differences in our nature and nurture, than that our brains have identical physico-chemical descriptions. (1978a, S. xvi)[2]

Der Punkt dieser Kritik ist, daß – so jedenfalls sieht es Dennett – der Funktionalismus die Auffassung impliziert, daß man zwei Wesen nur dann dieselben mentalen Zustände zuschreiben kann, wenn diese Wesen in gewisser Weise dieselbe funktionale Organisation aufweisen bzw. dasselbe Programm oder – um in der Terminologie Putnams zu bleiben – dieselbe Turing-Maschine realisieren. Und diese Auffassung ist Dennett zufolge einfach falsch.

In (1978b) erläutert Dennett diese Kritik am Beispiel eines Gerätes zur Erkennung und Einordnung von menschlichen Gesichtern – d.h. am Beispiel eines "face recognizer". Dabei geht Dennett von der Annahme aus, daß zwei Gruppen von Artificial Intelligence-Forschern den Auftrag erhalten, unabhängig voneinander je einen solchen "face recognizer" zu konstruieren. Was für ein Gerät können wir von jeder dieser beiden Gruppen erwarten? Aus ihrem Auftrag – d.h. allein schon aus dem Begriff des "face recognizer" – ergibt sich nach Dennett, daß

Erkenntnis **24** (1986) 309–341.

beide Gruppen versuchen werden, ein Gerät herzustellen, das nicht nur Fragen über ihm gegenübergestellte Gesichter richtig beantwortet, sondern das darüber hinaus seine Fähigkeit, Gesichter zu erkennen, auch in verschiedener Weise "verwenden" kann – je nach dem, was dieses Gerät sonst noch tut, welche Ziele es hat, . . . usw. Abgesehen davon aber, daß jeder "face recognizer" diese Fähigkeiten haben muß, sind der Art der Ausführung ansonsten keine Grenzen gesetzt.

At the physical level one might be electronic, the other hydraulic. Or one might rely on a digital computer, the other on an analogue computer. Or, at a higher level of design, one might use a system that analyzed exhibited faces via key features with indexed verbal labels – 'balding', 'snub-nosed', 'lantern-jawed' – and then compared label-scores against master lists of label-scores for previously encountered faces, while the other might use a system that reduced all face presentations to standard size and orientation, and checked them quasi-optically against stored 'templates' or 'stencils'. (1978a, S. 24; vgl. zu dieser Argumentation auch ein weiteres Beispiel in 1978a, S. 105).

Die von den beiden Gruppen angefertigten Geräte können sich in all diesen Punkten im Hinblick auf Material und Konstruktion voneinander unterscheiden und dennoch gleich gute – und zwar recht gute – "face recognizer" sein, insofern sie eben die für einen "face recognizer" erforderlichen Fähigkeiten besitzen. Ein "face recognizer" ist eben nicht durch eine bestimmte innere Struktur oder funktionale Organisation definiert, sondern nur dadurch, daß er bestimmte Fähigkeiten hat. Und in derselben Weise, so Dennett, ist auch ein Wesen mit Wünschen und Meinungen – also mit bestimmten mentalen Zuständen – nicht durch eine bestimmte funktionale Organisation, sondern nur durch die Fähigkeiten bestimmt, die für diese mentalen Zustände charakteristisch sind.

Diese Kritik Dennetts am Funktionalismus ist in gewisser Weise irritierend, da Dennetts Theorie selbst von vielen als eine Art Funktionalismus angesehen wird. In diesem Aufsatz möchte ich deshalb das Verhältnis zwischen Dennetts Intentionalismus und dem Funktionalismus Fodors und Putnams genauer untersuchen. Dabei, hoffe ich, wird klar werden, daß tatsächlich auch Dennetts Intentionalismus eine funktionalistische Theorie ist. Andererseits wird sich aber auch ergeben, daß man verschiedene Arten des Funktionalismus unterscheiden kann und daß Dennett eine sehr spezifische Form des Funktionalismus vertritt.

Zu Beginn scheint es mir jedoch sinnvoll zu sein, die Grundzüge der Theorie Dennetts noch einmal kurz zusammenzufassen. Dennett geht

in seinen Überlegungen aus von der Unterscheidung zwischen drei Erklärungsebenen bzw. Einstellungen ("stances"), die man seiner Meinung nach einnehmen kann, wenn man das Verhalten eines Systems erklären will: die *physikalische Einstellung* (physical stance), die "*funktionale*" *Einstellung* (design stance) und die *intentionale Einstellung* (intentional stance). In (1971) erläutert Dennett diese drei Begriffe – ebenso wie in (1973) – an seinem Lieblingsbeispiel: dem Verhalten eines Schachcomputers.

First there is the *design stance*. If one knows exactly how the computer is designed (including the impermanent part of its design: its program) one can predict its designed response to any move one makes by following the computation instructions of the program . . . a design of a system breaks it up into larger or smaller functional parts, and design-stance predictions are generated by assuming that each functional part will function properly. For instance, the radio engineer's schematic wiring diagrams have symbols for each resistor, capacitor, transistor, etc. – *each with its task to perform* – and he can give a design-stance prediction of the behavior of a circuit by assuming that each element performs its task. (1978a, S. 4; vgl. S. 237)

Die physikalische Einstellung dagegen charakterisiert Dennett so:

Second, there is what we may call the *physical stance*. From this stance our predictions are based on the actual state of the particular system, and are worked out by applying whatever knowledge we have of the laws of nature . . . . [In the case of a chess-playing computer] one could predict the response it would make in a chess game by tracing out the effects of the input energies all the way through the computer until once more type was pressed against paper and a response was printed. (1978a, S. 237)

Die dritte mögliche Einstellung – die intentionale Einstellung – nehmen wir Dennett zufolge insbesondere dann ein, wenn ein System so komplex ist, daß sein Verhalten mit Hilfe der anderen Einstellungen nicht mehr erklärt oder vorausgesagt werden kann. Bei einem Schachcomputer z.B. wäre der Versuch einer physikalischen Erklärung "eine witzlose Herkulesarbeit", die *praktisch* undurchführbar ist, obwohl sie *prinzipiell* natürlich durchgeführt werden könnte (vgl. unten Anm. 17). Wenn man einem System gegenüber die intentionale Einstellung einnimmt, dann versucht man das Verhalten dieses Systems zu erklären oder vorauszusagen, indem man dem System bestimmte Ziele und Informationen unterstellt und dann überlegt, welches Verhalten angesichts dieser Ziele und Informationen rational wäre, was also jemand mit diesen Zielen und Informationen über die gegebene Situation vernünftigerweise tun würde. Intentionale Erklärungen und Prognosen beruhen auf einer Antwort auf die Frage: "What is the most rational

thing for the computer to do, given the goals $x, y, z, \ldots$, constraints $a, b, c, \ldots$, and information $p, q, r, \ldots$" (1978a, p. 6). Anders ausgedrückt: intentionale Erklärungen beruhen darauf, daß man einem System Ziele und Überzeugungen im Hinblick auf die Situation, in der es sich befindet, zuschreibt und ihm zugleich Rationalität im Hinblick auf diese Ziele und Überzeugungen unterstellt.

> . . . [the] third stance . . . is the *intentional stance*; the predictions one makes from it are intentional predictions; one is viewing the computer as an intentional system. One predicts behavior in such a case by ascribing to the system *the possession of certain information* and supposing it to be *directed by certain goals*, and then by working out the most reasonable or appropriate action on the basis of these ascriptions and suppositions. It is a small step to calling the information possessed the computer's *beliefs*, its goals and subgoals its *desires* . . . . (1978a, S. 6)
>
> Whenever one can successfully adopt the intentional stance toward an object, I call that object an *intentional system* . . . . (1978a, S. 238)

Ein Objekt ist Dennett zufolge also genau dann ein intentionales System, wenn man sein Verhalten in intentionaler Einstellung – also durch die Zuschreibung von Zielen und Informationen (bzw. Wünschen und Meinungen) – erfolgreich erklären und voraussagen kann.

2.

Bei dem Versuch, das Verhältnis zwischen dem Intentionalismus Dennetts auf der einen und dem Funktionalismus insbesondere Putnams auf der anderen Seite zu klären, möchte ich von einer Unterscheidung ausgehen, die man bei der Erklärung des Verhaltens beliebiger Systeme[3] machen kann und die ebenso einfach wie grundlegend ist: der Unterscheidung zwischen internen und externen Verhaltenserklärungen. *Intern* sollen Verhaltenserklärungen heißen, die auf irgendeinem Wissen über die *innere Struktur* des betreffenden Systems beruhen, und *extern* Verhaltenserklärungen, bei denen das nicht der Fall ist.[4] Interne Verhaltenserklärungen setzen also voraus, daß die innere Struktur des Systems, dessen Verhalten erklärt werden soll, zumindest teilweise bekannt ist. Bei externen Verhaltenserklärungen dagegen ist dieses System für den Erklärenden eine *black box*.

Im Hinblick auf die gerade getroffene Unterscheidung ist zunächst klar, daß physikalische Erklärungen – also "physical stance"-Erklärungen im Sinne Dennetts – zu den internen Verhaltenserklärungen gehören. Dies scheint mir nicht problematisch zu sein, und ich werde auf diese Art von Erklärungen deshalb hier auch nicht weiter eingehen. Zu den internen Verhaltenserklärungen gehören aber auch die "design

stance"-Erklärungen Dennetts, was auf den ersten Blick vielleicht nicht ganz so selbstverständlich zu sein scheint. Aber die eigenen Aussagen Dennetts sind da ganz eindeutig:

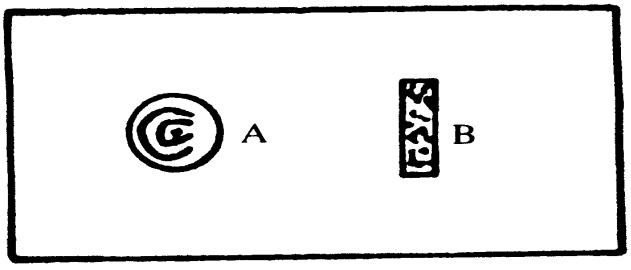
> . . . a design of a system breaks it up into larger or smaller functional parts, and design stance predictions are generated by assuming that each functional part will function properly. For instance, the radio engineer's schematic wiring diagrams have symbols for each resistor, capacitor, transistor, etc. – each with its task to perform – and he can give a design stance prediction of the behavior of a circuit by assuming that each element performs its task." (1978a, S. 4)

"Design stance"-Erklärungen und "design stance"-Prognosen beruhen, wie Dennett sie beschreibt, also darauf, daß man weiß, welche (funktionalen) Bausteine in einem System enthalten sind und wie diese Bausteine in dem System zusammenwirken. "Design stance"-Erklärungen sind also offenbar – und Dennetts Beispiel scheint mir da keinen Zweifel zu lassen – ihrer Natur nach Erklärungen aufgrund der Kenntnis von *Schaltplänen*.[5] Denn es sind gerade die Schaltpläne eines Systems, die angeben, welche (funktionalen) Bausteine dieses System enthält und wie diese Bausteine miteinander verdrahtet sind (bzw. allgemeiner: in welchen Beziehungen diese Bausteine zueinander stehen). Schaltpläne vermitteln genau das Wissen über die innere Struktur eines Systems, auf dem nach Dennett "design stance"-Erklärungen und "design stance"-Prognosen des Verhaltens dieses Systems beruhen. Bei Dennett selbst wird dies noch einmal ganz deutlich, wenn er in (1973) schreibt:

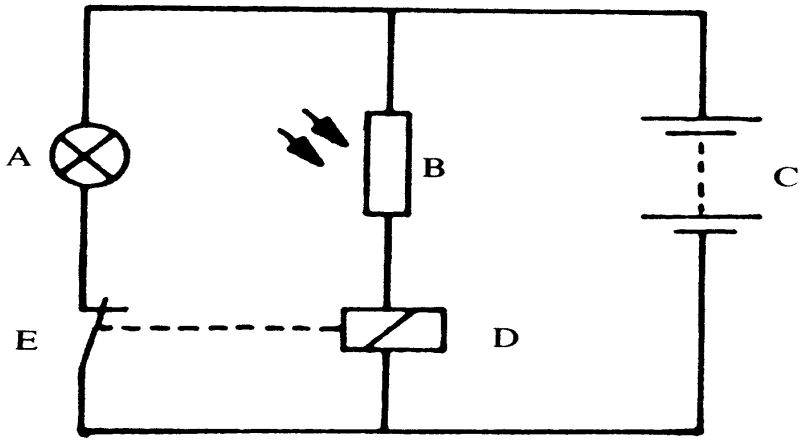
> In making a prediction from the design stance, one . . . predicts, as it were, from the *blueprints* alone. (1978a, S. 237; Hervorh. vom Verf.)

"Design stance"-Erklärungen und "design stance"-Prognosen könnte man mit einer gewissen Berechtigung also auch Schaltplan-Erklärungen und Schaltplan-Prognosen nennen.

Ich möchte mit Hilfe eines einfachen Beispiels versuchen, deutlicher zu machen, was mit diesen Ausdrücken gemeint ist. Ausgangspunkt soll ein einfaches System S1 sein, das von außen in etwa so aussieht:



Aus dem Aussehen des Systems S1 kann man noch so gut wie gar nichts über sein Verhalten schließen. Man kann vielleicht erkennen, daß es ein Lämpchen (A) und ein weiteres Bauteil (B) enthält, über das sonst nichts weiter bekannt ist; aber das reicht noch nicht aus, um etwas über das Verhalten von S1 aussagen zu können. Wenn man jedoch weiter erfährt, daß zu dem System S1 der folgende Schaltplan gehört, dann sieht die Sache schon anders aus:



Denn dieser Schaltplan zeigt erstens, welche funktionalen Bestandteile das System S1 enthält: ein Lämpchen A, einen lichtempfindlichen Widerstand B, eine Batterie C, ein Relais D und einen Schalter E. Und aus dem Schaltplan geht zweitens auch hervor, daß diese fünf Bauteile so miteinander verbunden sind, daß (a) das Relais D den Schalter E genau dann öffnet, wenn genügend Licht auf den Widerstand B trifft, und daß (b) das Lämpchen A genau dann leuchtet, wenn der Schalter E geschlossen ist. Aus dem angegebenen Schaltplan kann man also entnehmen, daß das Lämpchen A – vorausgesetzt alle Bauteile funktionieren wie vorgesehen – genau dann aufleuchtet, wenn sich S1 in einer relativ dunklen Umgebung befindet, bzw. genauer gesagt: wenn relativ wenig Licht auf das Bauteil B trifft.[6] Konkret kann man, falls man den Schaltplan von S1 kennt, also z.B. voraussagen, daß das Lämpchen von S1 nicht leuchtet, wenn man die Vorderseite dieses Systems mit einer Lampe anleuchtet oder das ganze System dem hellen Sonnenschein aussetzt, oder daß das Lämpchen von S1 aufleuchtet, wenn man das Bauteil B mit der Hand abdeckt oder das ganze System in einen nur sehr wenig beleuchteten Raum bringt.[7]

Soweit scheint also klar zu sein, was Schaltplan-Erklärungen sind und inwiefern man die "design stance"-Erklärungen Dennetts in diesem Sinne als Schaltplan-Erklärungen deuten kann. Auf der anderen Seite fällt aber auch auf, daß Dennett "design stance"-Erklärungen immer wieder mit dem Begriff des Programms bzw. mit der Möglichkeit der Erklärung durch Programme in Verbindung bringt. Dies wird schon klar an der Weise, in der er den Begriff der "design stance"-Erklärung in "Intentional Systems" erläutert (vgl. oben Abschn. 1). Noch prägnanter äußert er sich aber in (1973).

> First there is the design stance. If one knows exactly how the computer's *program* has been designed (. . .), one can predict the computer's designed response to any move one makes. (1978a, S. 237; Hervorh. vom Verf.)

Neben der Möglichkeit, "design stance"-Erklärungen als Schaltplan-Erklärungen zu interpretieren, scheint es nach den Aussagen Dennetts also auch die Möglichkeit zu geben, diese Erklärungen als Erklärungen durch Programme (bzw. kurz: als Programm-Erklärungen) aufzufassen. Dennett selbst macht da allerdings – wenn überhaupt – keinen großen Unterschied. Meiner Meinung nach ist der Unterschied zwischen Schaltplan- und Programm-Erklärungen jedoch keineswegs unerheblich. Auf einen einfachen Nenner gebracht besteht dieser Unterschied darin, daß bei Programm-Erklärungen das Verhalten (der "output") eines Systems nicht dadurch erklärt wird, daß man das System "in größere oder kleinere funktionale Bestandteile aufbricht", sondern dadurch, daß man angibt, wie das System diesen output bei gegebenem input *sequentiell* – d.h. in einer Reihe von zeitlich aufeinander folgenden endlich vielen diskreten Schritten – erzeugt, wobei über die Bestandteile des Systems gar nichts weiter ausgesagt wird. Dies wird insbesondere auch deutlich, wenn man sich klar macht, daß ein Programm nichts anderes ist als eine endliche Folge von Anweisungen, denen im Computer jeweils ein einzelner – mehr oder weniger großer – Arbeitsschritt entspricht. Daß das Verhalten eines Computers durch sein Programm bestimmt ist, bedeutet daher, daß in diesem Computer bei gegebenem input ein bestimmter output erzeugt wird, indem der Computer die einzelnen Arbeitsschritte der Erzeugung in der durch das Programm festgelegten Reihenfolge ausführt.

Auch hier kann vielleicht ein einfaches Beispiel zu einem besseren Verständnis helfen. Denken wir uns ein System S2, in das über eine Tastatur positive ganze Zahlen eingegeben werden können und das bei

jeder eingegebenen Zahl entweder eine Eins oder eine Null ausdruckt. Dabei soll zunächst noch nicht bekannt sein, bei welchen Zahlen eine Null und bei welchen eine Eins ausgegeben wird. Auch bei diesem System hilft uns das äußere Aussehen nicht sehr viel, wenn es darum geht, für eine bestimmte eingegebene Zahl vorauszusagen, ob S2 eine Null oder eine Eins ausdrucken wird. Dagegen können wir in diesem Fall zu einer klaren Voraussage kommen, wenn wir erfahren, daß das System S2 nach dem folgenden Programm arbeitet.[8]

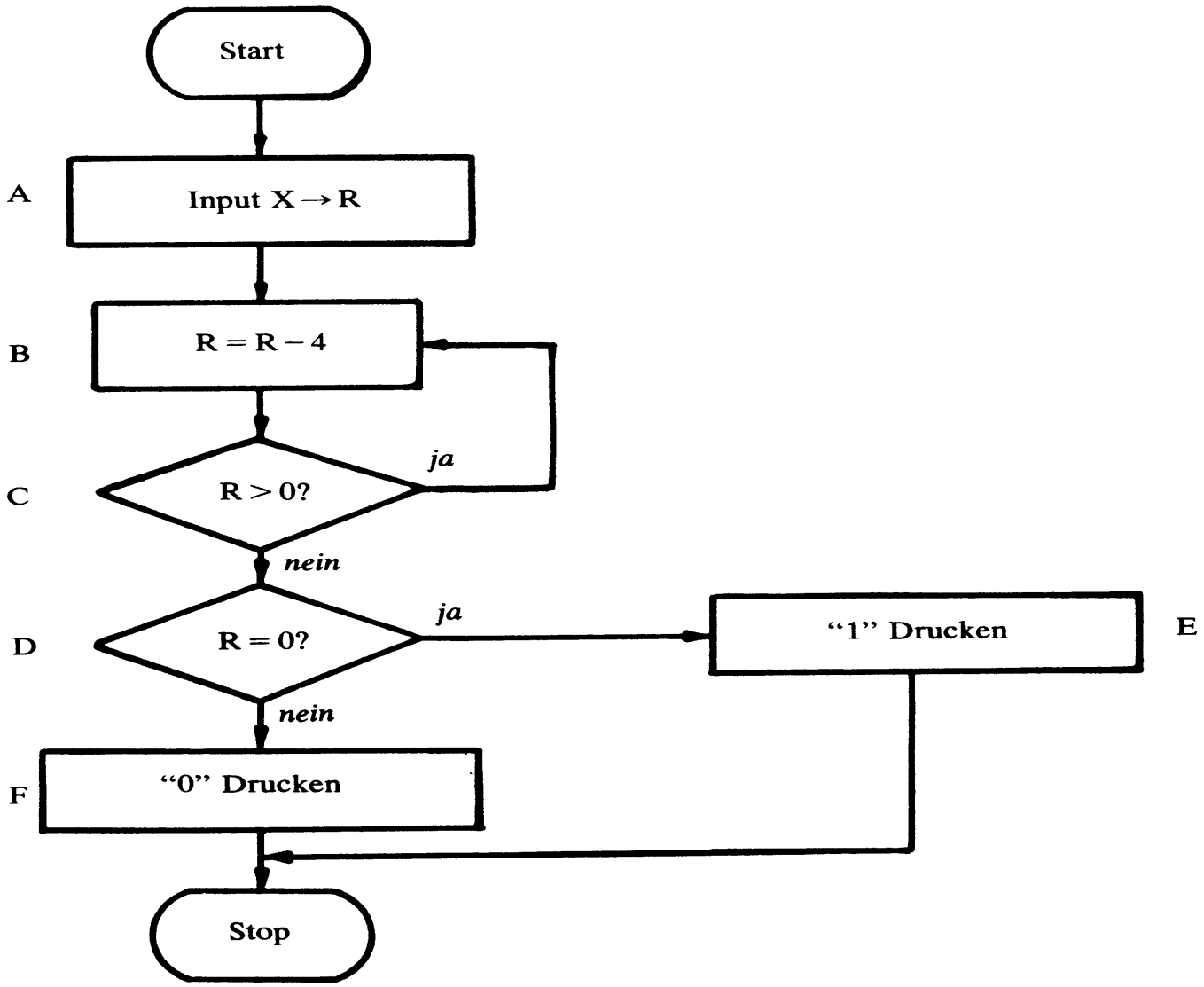
```
PS2    10:  INPUT A
       20:  A = A − 4
       30:  IF A > 0 THEN 20 ELSE 40
       40:  IF A = 0 THEN PRINT "1" ELSE PRINT "0"
       50:  END
```

Dieses Programm bedeutet für die Arbeitsweise von S2 folgendes: Im ersten Schritt (Anweisung "10") wird die eingegebene Zahl – sagen wir x – dem Register (der Variablen) A zugewiesen, im zweiten Schritt (Anweisung "20") wird der Inhalt von A um 4 vermindert. Im dritten Schritt (Anweisung "30") wird überprüft, ob der (neue) Inhalt von A größer als Null ist; ist das der Fall, so geht S2 zur Anweisung "20" zurück und vermindert im nächsten Schritt den Inhalt von A nochmals um 4. Im folgenden Schritt wird wieder geprüft, ob der Inhalt von A jetzt immer noch größer als Null ist; falls ja, geht S2 wieder zur Anweisung "20" . . . usw., bis der Inhalt von A nicht mehr größer als Null ist. Ist das erreicht, wird im nächsten Schritt (Anweisung "40") geprüft, ob der Inhalt von A jetzt gleich Null ist. Ist das der Fall, druckt S2 eine Eins und beendet dann (Anweisung "50") seine Arbeit; ist das nicht der Fall, druckt S2 eine Null und beendet ebenfalls (Anweisung "50") seine Arbeit. Damit ist der Arbeitsgang beendet, und S2 hat je nach eingegebener Zahl eine Eins oder eine Null ausgedruckt.

Zusammenfassend läßt sich die durch das Programm PS2 vorgegebene Arbeitsweise von S2 so darstellen: die in S2 eingegebene Zahl wird solange um 4 vermindert, bis das Ergebnis nicht mehr größer als Null ist; ist das Ergebnis dann gleich Null, druckt S2 eine Eins, ist es kleiner als Null, eine Null. Kurz, S2 arbeitet so, daß es eine Eins druckt, wenn die eingegebene Zahl durch 4 teilbar ist, und eine Null, wenn das nicht der Fall ist. Wenn man das Programm von S2 kennt, kann man also z.B. voraussagen, daß S2 eine Eins drucken wird, wenn man die Zahl 52 eingibt, bzw. eine Null, wenn die eingegebene Zahl 13 lautet.

Die Arbeitsweise programmgesteuerter Maschinen, zu denen auch das System S2 gehört, läßt sich nicht nur durch Programme – wie das angeführte Programm PS2 – sondern auch durch Flußdiagramme darstellen. Denn Flußdiagramme geben ebenso wie Programme an, welche Arbeitsschritte ein System in welcher Reihenfolge durchführt. Flußdiagramme und Programme sind insofern in gewisser Weise äquivalent. So kann man das Verhalten von S2 z.B. auch durch das folgende Flußdiagramm charakterisieren:

FS2



Auf der anderen Seite sind Flußdiagramme jedoch allgemeiner als Programme. Denn sie benutzen noch keine spezielle Programmiersprache, und an ihnen wird auch noch nicht deutlich, ob die Steuerung der durch das Flußdiagramm vorgegebenen Arbeitsweise des Systems tatsächlich im engeren Sinne programmiert oder durch feste Verdrahtung oder sogar "nur" elektromechanisch oder mechanisch realisiert ist (wie z.B. bei älteren Druckmaschinen).

An Flußdiagrammen wird auch in besonderer Weise klar, wie sich Programm-Erklärungen von Schaltplan-Erklärungen unterscheiden. Obwohl auf den ersten Blick eine große Ähnlichkeit zu bestehen scheint, sind Flußdiagramme nämlich der Art nach etwas ganz anderes als Schaltpläne. Denn die einzelnen Kästchen in einem Flußdiagramm stehen nicht für bestimmte funktional charakterisierte Bestandteile des Systems, sondern für bestimmte Arbeitsschritte. Ebenso wie Programme geben also auch Flußdiagramme nicht an, welche (funktionalen) Bestandteile ein System enthält und wie diese Bestandteile zusammenwirken. Sie sagen vielmehr etwas aus über die zeitliche Struktur der Arbeitsweise: die Reihenfolge, in der die einzelnen durch die Kästchen des Flußdiagramms repräsentierten Arbeitsschritte ausgeführt werden. Noch genauer kann man sagen, daß die einzelnen Kästchen in einem Flußdiagramm nicht funktionale Bestandteile, sondern *funktionale Zustände* eines Systems symbolisieren. Die Inschriften der Kästchen geben an, was das System "tut", wenn es in dem betreffenden Zustand ist, und die Pfeile geben an, wie die einzelnen Zustände miteinander zusammenhängen. In dem angegebenen Flußdiagramm z.B. kann man die verschiedenen Kästchen mit sechs verschiedenen Zuständen A–F identifizieren, die das System S2 annehmen kann und die dieses System in einer bestimmten Reihenfolge durchläuft. Einzeln kann man diese Zustände wie folgt charakterisieren:

A Wenn S2 im Zustand A ist, weist es eine eingegebene Zahl X dem Register R zu und geht dann in den Zustand B über.

B Wenn S2 im Zustand B ist, vermindert S2 den Inhalt von Register R um 4 und geht dann in den Zustand C über.

C Wenn S2 im Zustand C ist, geht es in den Zustand B über, falls der Inhalt von Register R größer als Null ist; sonst in den Zustand D.

D Wenn S2 im Zustand D ist, geht es in den Zustand E über,

falls der Inhalt von Register R gleich Null ist; sonst in den Zustand F.

E Wenn S2 im Zustand E ist, druckt S2 eine "1" und stoppt dann.

F Wenn S2 im Zustand F ist, druckt S2 eine "0" und stoppt dann.

Generell kann man Schaltplan-Erklärungen also als *Erklärungen durch funktionale Bauteile*, Programm-Erklärungen dagegen als *Erklärungen durch funktionale Zustände* charakterisiereren. Daß Dennett zwischen diesen beiden Arten der Erklärung nicht unterscheidet, da er sie beide einfach als "design stance"-Erklärungen kennzeichnet, ist sicher mit ein Grund für sein ungeklärtes Verhältnis zum Funktionalismus. Aber darüber wird später noch mehr zu sagen sein. Im übrigen bedeutet die Tatsache, daß man Schaltplan- und Programm-Erklärungen voneinander unterscheiden sollte, jedoch nicht, daß es nicht auch Mischformen zwischen diesen beiden Arten der Erklärung geben könnte – Mischformen, die z.B. dann auftreten, wenn in einem sequentiell arbeitenden System die verschiedenen funktionalen Bestandteile in einer bestimmten Reihenfolge angesprochen werden. Auf diese Mischformen will ich jedoch hier auch nicht weiter eingehen.

3.

Nach den internen Verhaltenserklärungen sollen in diesem Abschnitt nun die externen Verhaltenserklärungen etwas weiter erläutert werden. Dabei läßt sich das Prinzip dieser Art von Erklärungen wieder recht gut am Beispiel des oben angeführten Systems S1 verdeutlichen. Die Ausgangsfrage lautet jetzt aber: wie läßt sich das Verhalten von S1 erklären oder voraussagen, wenn man *weder* den Schaltplan dieses Systems kennt *noch sonst in der Lage ist, irgendetwas über den inneren Aufbau dieses Systems herauszufinden*. Offenbar gibt es auf diese Frage nur eine Antwort: man muß das Verhalten von S1 über einen längeren Zeitraum hinweg beobachten und aus diesen Beobachtungen dann die entsprechenden Schlüsse ziehen.

Wenn man bei dem System S1 auf diese Weise vorgeht, dann wird man z.B. feststellen (vgl. oben Abschn. 2), daß das Lämpchen von S1 nicht leuchtet, wenn man das System mit einer Lampe anstrahlt oder

der hellen Sonne aussetzt, während es auf der anderen Seite leuchtet, wenn man S1 in einen dunklen Raum bringt oder die Vorderseite von S1 mit der Hand abdeckt. Aufgrund von Beobachtungen kann man im Laufe der Zeit die folgenden Feststellungen treffen:

(3.1) Wenn man S1 dem hellen Sonnenlicht aussetzt, dann leuchtet das Lämpchen A nicht.

(3.2) Wenn man die Vorderseite von S1 mit einer Lampe anstrahlt, dann leuchtet das Lämpchen A nicht.

(3.3) Wenn man S1 in einen dunklen Raum bringt, leuchtet das Lämpchen A.

(3.4) Wenn man die Vorderseite von S1 abdeckt, leuchtet das Lämpchen A.

Die in diesen Aussagen festgehaltenen Beobachtungen zeigen zunächst nur, daß das Leuchten des Lämpchens von S1 davon abhängt, wieviel Licht auf das System fällt. Durch genauere Beobachtungen läßt sich aber weiter eingrenzen, daß es letzten Endes nur darauf ankommt, wieviel Licht auf das Bauteil B fällt. Und durch systematische Variierung der Lichtquelle läßt sich sogar feststellen, wieviel Licht mindestens auf dieses Bauteil fallen muß, damit das Lämpchen A zu leuchten aufhört. Genaue Beobachtungen ermöglichen es also, schließlich die folgende gesetzesartige Aussage zu formulieren:

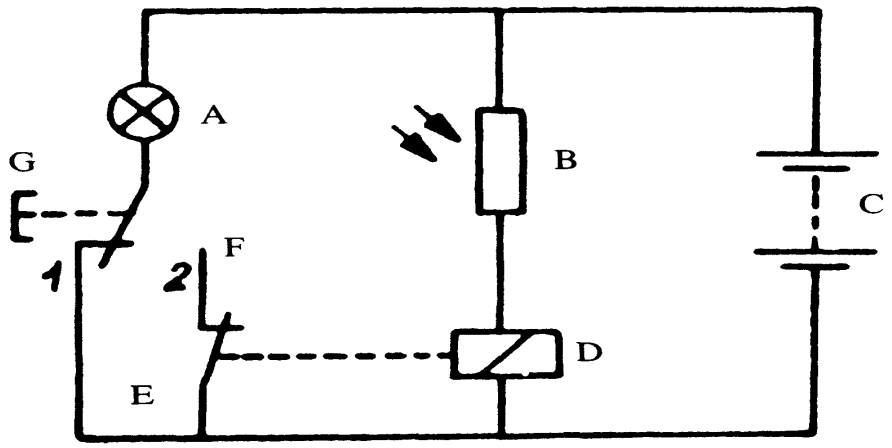
(3.5) Das Lämpchen von S1 leuchtet – unter normalen Umständen – genau dann, wenn auf das Bauteil B von S1 weniger als A Lux Licht fallen.

Wenn man diese gesetzesartige Aussage behauptet, dann schreibt man S1 aber eine bestimmte Disposition zu. Und wenn man das Verhalten von S1 unter Bezugnahme auf diese Aussage erklärt, dann gibt man auf diese Weise eine dispositionelle Erklärung für das Verhalten dieses Systems. Der Begriff der *Disposition* ist also der Zentralbegriff der externen Verhaltenserklärungen, und ihrer Struktur nach können diese Erklärungen somit als dispositionelle Erklärungen aufgefaßt werden.

Zu den dispositionellen Erklärungen im weiteren Sinne gehört auch ein Typ von Erklärungen, die ich aus Gründen, von denen ich hoffe, daß sie im folgenden deutlich werden, "theoretisch-funktionale" Erklärungen nennen möchte (bzw. kurz: TF-Erklärungen). Erklärungen

dieses Typs sind immer dann erforderlich, wenn ein System, dessen Verhalten nur extern erklärt werden kann, da über seine innere Struktur nichts bekannt ist, in dem Sinne ein *uneinheitliches* Verhalten zeigt, daß es in *qualitativ gleichen Situationen unterschiedlich* reagiert. Denn wenn dies der Fall ist, läßt sich das Verhalten des betreffenden Systems nicht mehr auf eine einfache Disposition zurückführen.

Ein weiteres Beispiel soll hier wieder zur Verdeutlichung dienen. Ausgangspunkt sei dieses Mal ein System S1*, das dem System S1 von außen sehr ähnlich sieht, an dem man jedoch äußerlich außer dem Lämpchen A und dem Bauteil B auch noch einen Druckknopf G bemerken kann. Diesem System soll der folgende Schaltplan entsprechen:



Aus diesem Schaltplan ergibt sich, daß das System S1* unter anderem die gleichen Bauteile enthält wie das System S1: ein Lämpchen A, einen lichtempfindlichen Widerstand B, eine Batterie C, ein Relais D und einen Schalter E. Zusätzlich enthält S1* jedoch auch noch einen Umschalter F und einen Druckknopf G, mit dem der Umschalter F durch wiederholtes Drücken von der Position 1 in die Position 2 und umgekehrt gebracht werden kann. Beim ersten Drücken von G z.B. wird das Schaltglied 1 geöffnet und zugleich das Schaltglied 2 geschlossen, beim nächsten Drücken dann das Schaltglied 1 geschlossen und das Schaltglied 2 geöffnet, beim nächsten Drücken wieder das Schaltglied 1 geöffnet und das Schaltglied 2 geschlossen, . . . usw.

Für die Arbeitsweise von S1* bedeutet das folgendes: wenn das Schaltglied 1 geschlossen und das Schaltglied 2 geöffnet ist, dann ist der

Stromkreis, der das Lämpchen A mit der Batterie C verbindet, geschlossen; in diesem Fall leuchtet das Lämpchen A also durchgehend. Ist dagegen das Schaltglied 1 geöffnet und das Schaltglied 2 geschlossen, dann verhält sich S1* genauso wie S1, d.h. in diesem Fall leuchtet das Lämpchen von S1* genau dann, wenn relativ wenig Licht auf den Widerstand B fällt.

Das durch die gerade erläuterte Schaltung bedingte (uneinheitliche) Verhalten von S1* ist natürlich *auch ohne Kenntnis des entsprechenden Schaltplans* festzustellen, wenn man S1* über einen längeren Zeitraum hinweg beobachtet. *Manchmal* leuchtet das Lämpchen von S1* durchgehend – unabhängig davon, welche Bedingungen ansonsten gegeben sind. *Manchmal* jedoch hört das Lämpchen zu leuchten auf, wenn man S1* in eine helle Umgebung bringt, und beginnt wieder zu leuchten, wenn man die Vorderseite des Systems mit der Hand abdeckt. Und dabei kommt es offensichtlich darauf an, ob und wann der Druckknopf G betätigt wird. Diese Beobachtungen über das Verhalten von S1* lassen sich, wie gesagt, nicht mehr auf eine einfache Disposition zurückführen; aber sie lassen sich erklären, *wenn man annimmt, daß das System S1* zwei verschiedene Zustände Z1 und Z2 annehmen kann, denen gewissermaßen zwei verschiedene Dispositionen entsprechen, oder* – um es präziser auszudrücken – *die durch die folgenden gesetzesartigen Aussagen charakterisiert sind*:

(3.6) Wenn S1* im Zustand Z1 ist, dann leuchtet das Lämpchen von S1* durchgehend.

(3.7) Wenn S1* im Zustand Z2 ist, dann leuchtet das Lämpchen von S1* genau dann, wenn auf das Bauteil B weniger als A Lux Licht fallen.

(3.8) Wenn S1* im Zustand Z1 ist, dann geht S1* genau dann in den Zustand Z2 über, wenn der Knopf G gedrückt wird.

(3.9) Wenn S1* im Zustand Z2 ist, dann geht S1* genau dann in den Zustand Z1 über, wenn der Knopf G gedrückt wird.

Mit Hilfe dieser vier Aussagen kann man Prognosen über das Verhalten von S1* aufstellen – zumindest, wenn man den entsprechenden Anfangszustand von S1* kennt (was aber offensichtlich leicht zu erreichen ist). Wenn bekannt ist, daß sich S1* im Zustand Z1 befindet, kann man z.B. voraussagen, daß das Lämpchen A nicht leuchten wird, wenn man

den Knopf G drückt und S1* mit einer Taschenlampe anleuchtet; und wenn bekannt ist, daß sich S1* im Zustand Z2 befindet, kann man z.B. voraussagen, daß das Lämpchen A, falls man den Knopf G drückt, auch dann leuchtet, wenn man S1* dem hellen Sonnenlicht aussetzt. In analoger Weise kann man auf die Aussagen (3.6)–(3.9) auch Erklärungen des Verhaltens von S1* stützen – z.B. die Erklärung "Das Lämpchen von S1* hat gestern um 18.30 Uhr geleuchtet, da sich S1* zu diesem Zeitpunkt im Zustand Z1 befand" oder die Erklärung "Das Lämpchen von S1* hat vor einer halben Stunde nicht geleuchtet, weil S1* sich zu diesem Zeitpunkt im Zustand Z2 befand und einer hellen Lichtquelle ausgesetzt war".

Das Fazit des gerade erläuterten Beispiels scheint mir in folgendem zu liegen. In TF-Erklärungen wird das Verhalten eines Systems dadurch erklärt, daß man unterstellt, daß dieses System verschiedene "innere" Zustände annehmen kann, denen jeweils eine andere Verhaltensdisposition entspricht. Insofern könnte es zunächst naheliegen, TF-Erklärungen zu den internen Verhaltenserklärungen zu zählen. Zumal eine solche Tendenz auch noch durch den Umstand gestützt wird, daß es offensichtlich eine starke Verwandtschaft zwischen TF-Erklärungen auf der einen Seite und Programm-Erklärungen durch funktionale Zustände auf der anderen Seite gibt. Dennoch wäre eine solche Interpretation irreführend. Denn TF-Erklärungen beruhen – ebenso wie einfache dispositionelle Erklärungen – *nicht auf einem vorab verfügbaren Wissen über die innere Struktur des Systems*, dessen Verhalten erklärt werden soll; sie *unterstellen* dem System vielmehr eine Art von "innerer" Struktur – eine Menge von nicht ohne weiteres beobachtbaren Zuständen, die zueinander und zu den möglichen inputs und outputs des Systems in bestimmten Beziehungen stehen –, da nur so das Verhalten des Systems sinnvoll erklärt werden kann. Die Zustände, auf die in TF-Erklärungen Bezug genommen wird, sind insofern auch nur durch die gesetzesartigen Aussagen charakterisiert, in denen sie vorkommen – so wie die Zustände Z1 und Z2 von S1* nur durch die Aussagen (3.6)–(3.9) bestimmt sind. Mehr ist von ihnen (vorerst) nicht bekannt. Damit, hoffe ich, ist jetzt auch klar, warum ich Erklärungen dieser Art "theoretisch-funktionale" Erklärungen nennen möchte. Denn die Zustände, auf die sich diese Erklärungen beziehen, sind als *theoretische* Zustände anzusehen, weil ihre Existenz *postuliert* wird, *da nur so eine befriedigende Erklärung des Verhaltens der entsprechenden Systeme möglich ist*. Und als *funktionale* Zustände müssen sie gelten, *da*

sie nur durch die Rolle (oder Funktion) charakterisiert sind, die sie bei der Hervorbringung dieses Verhaltens spielen.

4.

Im letzten Abschnitt sind TF-Erklärungen nur relativ kurz und in ihren Grundzügen im Zusammenhang mit dem Beispiel des Systems S1* erläutert worden. In diesem Abschnitt sollen deshalb noch einige weitere, bisher nicht angesprochene Punkte zur Sprache kommen, die für das Verständnis dieser Erklärungen wichtig sind. Der erste Punkt, auf den ich hier kurz eingehen möchte, läßt sich folgendermaßen erläutern.

TF-Erklärungen beruhen, wie ich zu zeigen versucht habe, immer auf einer Theorie – einer Theorie, wie sie z.B. in der vier Verhaltensgesetzen (3.6)–(3.9) für das System S1* formuliert ist. Eine solche Theorie kann jedoch nicht nur dazu verwendet werden, das Verhalten eines Systems zu erklären, sie kann zugleich auch dazu dienen, den Typ des entsprechenden Systems zu definieren. Und dies gilt in analoger Weise auch für die theoretisch-funktionalen Zustände, die dieses System annehmen kann. Bei derartigen Definitionen geht man jedoch nicht von der jeweiligen Theorie selbst, sondern vom Ramsey-Satz dieser Theorie aus. Damit ist folgendes gemeint.[9] Sei z.B. T eine Theorie, in der das Verhalten eines Systems S erklärt wird, indem diesem System die (theoretischen) Zustände $t_1, \ldots, t_n$ zugeschrieben werden – eine solche Theorie soll im folgenden kurz mit $T(S, t_1, \ldots, t_n)$ bezeichnet werden –, dann erhält man den Ramsey-Satz dieser Theorie, indem man in ihr alle Vorkommnisse der Ausdrücke $t_1, \ldots, t_n$ durch entsprechende freie Variablen $x_1, \ldots, x_n$ ersetzt und zugleich diese Variablen durch Existenzquantoren bindet. Der Ramsey-Satz von T lautet dann in formaler Kurzschreibweise:

(4.1) $\quad \mathrm{V}x_1 \ldots \mathrm{V}x_n \mathrm{T}(\mathrm{S}, x_1, \ldots, x_n).$[10]

Wenn man nun den Typ von S definieren will, muß man in diesem Ramsey-Satz auch noch den Term S durch eine Variable – sagen wir x – ersetzen, so daß man zu der Satzfunktion kommt:

(4.2) $\quad \mathrm{V}x_1 \ldots \mathrm{V}x_n \mathrm{T}(x, x_1, \ldots, x_n).$

Mit Hilfe dieser Satzfunktion kann man den Typ von S dann durch die Festlegung definieren:

(4.3) x ist ein System vom Typ S genau dann, wenn gilt:
$\mathrm{V}x_1 \ldots \mathrm{V}x_n\mathrm{T}(x, x_1, \ldots, x_n)$.

In analoger Weise lassen sich ausgehend von der Satzfunktion (4.2) in einem zweiten Schritt die (theoretischen) Ausdrücke $t_1, \ldots, t_n$ genauer bestimmen. So kann man z.B. präzise definieren, was es heißt, daß sich ein System X im Zustand t_1 befindet, indem man festlegt:

(4.4) X ist im Zustand t_1 genau dann, wenn gilt:
$\mathrm{V}x_1 \ldots \mathrm{V}x_n(\mathrm{T}(X, x_1, \ldots, x_n)$ und X ist in $x_1)$.

Definitionen für System-Typen oder theoretisch-funktionale Zustände wie die Definitionen (4.3) und (4.4) sollen im folgenden *funktionale Definitionen* genannt werden.

Mit Hilfe des Systems S1* lassen sich die gerade geschilderten Verfahrensweisen sehr gut veranschaulichen. Wenn man Systeme wie S1* z.B. als "umschaltbare Helligkeitsanzeiger" bezeichnet, dann läßt sich dieser Ausdruck in der gerade geschilderten Weise präzise definieren. Die Grundidee ist dabei, daß etwas genau dann ein "umschaltbarer Helligkeitsanzeiger" sein soll, wenn es die gleiche *äußere Form* wie S1* hat und das *gleiche Verhalten* wie S1* zeigt. D.h. ein beliebiges System soll genau dann ein "umschaltbarer Helligkeitsanzeiger" sein, wenn es ein Lämpchen, ein nicht näher bestimmtes Bauteil B und einen Druckknopf enthält, so daß folgendes gilt: manchmal leuchtet das Lämpchen des Systems durchgehend; nach einmaligem Betätigen des Druckknopfes leuchtet es jedoch nur noch, wenn relativ wenig Licht auf das Bauteil B fällt; nach nochmaligem Betätigen des Druckknopfes leuchtet es wieder durchgehend . . . usw. Diese Grundidee läßt sich in der geschilderten Weise formal korrekt umsetzen. Ausgangspunkt sind dabei die vier Verhaltensgesetze (3.6)–(3.9), in denen zur Erklärung des Verhaltens von S1* die beiden Zustände Z1 und Z2 eingeführt wurden. Die Konjunktion dieser vier Aussagen soll im folgenden als Theorie T* bezeichnet werden. Im ersten Schritt wird nun zunächst der Ramsey-Satz $\mathrm{V}x_1\mathrm{V}x_2\mathrm{T}^*(\mathrm{S1}^*, x_1, x_2)$ dieser Theorie gebildet, und im zweiten Schritt wird in diesem Ramsey-Satz auch noch der Term S1* durch eine Variable ersetzt. Auf diese Weise erhält man die Satzfunktion:

(4.5) $\mathrm{V}x_1\mathrm{V}x_2$ (Wenn x in x_1 ist, dann leuchtet das Lämpchen von x durchgehend, und wenn x in x_2 ist, dann leuchtet das Lämpchen von x genau dann, wenn auf das Bauteil B

weniger als A Lux Licht fallen, und wenn x in x_1 ist, dann geht x genau dann in x_2 über, wenn der Knopf G gedrückt wird, und wenn x in x_2 ist, dann geht x genau dann in x_1 über, wenn der Knopf G gedrückt wird)

Wenn man diese Satzfunktion mit $\mathrm{V}x_1\mathrm{V}x_2\mathrm{T}^*(x, x_1, x_2)$ abkürzt, kann man den Ausdruck "umschaltbarer Helligkeitsanzeiger" nun präzise so definieren:

(4.6) x ist ein umschaltbarer Helligkeitsanzeiger genau dann, wenn gilt:
$\mathrm{V}x_1\mathrm{V}x_2\mathrm{T}^*(x, x_1, x_2)$.

Und weiter kann man auf diese Weise auch die beiden theoretisch-funktionalen Zustände von S1* genauer charakterisieren. Denn wenn man den Zustand von S1*, in dem das Lämpchen A ständig leuchtet, als "Dauerleuchten" und den zweiten theoretisch-funktionalen Zustand von S1* als "Helligkeitsanzeige" bezeichnet, dann lassen sich – ausgehend wiederum von der Satzfunktion $\mathrm{V}x_1\mathrm{V}x_2\mathrm{T}^*(x, x_1, x_2)$ – diese beiden Ausdrücke so definieren:

(4.7) x ist im Zustand Dauerleuchten genau dann, wenn gilt:
$\mathrm{V}x_1\mathrm{V}x_2(\mathrm{T}^*(x, x_1, x_2)$ und x ist in $x_1)$.

(4.8) x ist im Zustand Helligkeitsanzeige genau dann, wenn gilt:
$\mathrm{V}x_1\mathrm{V}x_2(\mathrm{T}^*(x, x_1, x_2)$ und x ist in $x_2)$.[11]

Die gerade geschilderten Überlegungen sind auch im Hinblick auf die viel umstrittene Frage relevant, wie sich die in TF-Erklärungen *postulierten* theoretisch-funktionalen Zustände zu den "*wirklichen*" physikalischen Zuständen eines Systems verhalten, ob sie von diesen physikalischen Zuständen unabhängig sind oder ob sie sich in irgendeiner Weise auf diese Zustände reduzieren oder sogar mit ihnen identifizieren lassen.

Das Problem, das mit dieser Frage angesprochen ist, läßt sich wieder sehr gut anhand des Systems S1* erläutern. Die theoretisch-funktionalen Zustände dieses Systems sind, wie schon gesagt, allein durch die vier Aussagen (3.6)–(3.9) der Theorie T* charakterisiert. Wenn man nun den Schaltplan von S1* noch einmal betrachtet, dann zeigt sich, daß sich am System S1* auch zwei *physikalische* Zustände[12] unterscheiden lassen – nämlich die beiden Zustände "Umschalter F ist in Position 1 (Schaltglied 1 geschlossen/Schaltglied 2 offen)" und

"Umschalter F ist in Position 2 (Schaltglied 1 geöffnet/Schaltglied 2 geschlossen)" –, auf die genau das zutrifft, was in der Theorie T* von den Zuständen Z1 und Z2 behauptet wird: wenn S1* im Zustand "Umschalter F ist in Position 1" ist, dann leuchtet das Lämpchen A durchgehend; wenn S1* im Zustand "Umschalter F ist in Position 2" ist, dann leuchtet das Lämpchen A genau dann, wenn relativ wenig Licht auf das Bauteil B fällt; durch Betätigen des Druckknopfes G geht S1* vom Zustand "Umschalter F ist in Position 1" in den Zustand "Umschalter F ist in Position 2" über und umgekehrt. Das Zustandspaar ("Umschalter F ist in Position 1", "Umschalter F ist in Position 2") erfüllt die folgenden vier Satzfunktionen also genauso wie das Zustandspaar (Z1, Z2) (bzw. wie das Zustandspaar (Dauerleuchten, Helligkeitsanzeige)):

(3.6′) Wenn S1* in x_1 ist, dann leuchtet das Lämpchen von S1* durchgehend.

(3.7′) Wenn S1* in x_2 ist, dann leuchtet das Lämpchen von S1* genau dann, wenn auf das Bauteil B weniger als A Lux Licht fallen.

(3.8′) Wenn S1* in x_1 ist, dann geht S1* genau dann in x_2 über, wenn der Knopf G gedrückt wird.

(3.9′) Wenn S1* in x_2 ist, dann geht S1* genau dann in x_1 über, wenn der Knopf G gedrückt wird.

Angesichts dieser Tatsache liegt aber die Frage nahe, ob es sich hier wirklich um verschiedene Zustände handelt oder ob es nicht sinnvoller ist anzunehmen, daß die beiden genannten physikalischen Zustände von S1* – sie sollen im folgenden mit P1 und P2 bezeichnet werden – mit den theoretisch-funktionalen Zuständen Z1 und Z2 identifiziert werden können, so daß man z.B. sagen kann: der theoretisch-funktionale Zustand Z1 bzw. Dauerleuchten *ist identisch* mit dem Zustand "Umschalter F ist in Position 1"; oder sogar: der Zustand Z1 *ist* der Zustand "Umschalter F ist in Position 1".

Meiner Meinung nach können die vorangegangenen Überlegungen zur Beantwortung dieser Frage folgendes beitragen: Aus der Definition (4.7) ergibt sich z.B., daß S1* genau dann im Zustand "Dauerleuchten" ist, wenn es zwei Zustände gibt, die die Satzfunktion $T^*(S1^*, x_1, x_2)$ erfüllen, und wenn S1* im ersten dieser beiden Zustände ist. Zumindest

für den Fall, daß das Paar (P1, P2) das einzige Paar von Zuständen ist, das diese Satzfunktion erfüllt,[13] liegt es also nahe, den Zustand P1 mit dem Zustand "Dauerleuchten" zu identifizieren (und analog gilt dies auch für P2 und den Zustand "Helligkeitsanzeige"). Denn in diesem Fall ist P1 und nur P1 der Zustand, der zusammen mit P2 die genannte Satzfunktion erfüllt, so daß in diesem Fall die Aussage "S1* ist im Zustand Dauerleuchten" dann und nur dann wahr ist, wenn die Aussage "S1* ist im Zustand P1" wahr ist. Diese Tatsache allein reicht jedoch – obwohl sie an sich ein starkes Indiz darstellt – noch nicht aus, um auf die Identität von P1 und Z1 schließen zu können. Der Grund für diese Schwierigkeit scheint mir letztlich darin zu liegen, daß in der Definition (4.7) nur der Ausdruck "x befindet sich im Zustand Dauerleuchten", aber nicht der Ausdruck "Z ist im System x der Zustand Dauerleuchten" definiert wird. Es liegt jedoch nahe, den Ausdruck "Dauerleuchten" in dem Sinne als eine *Kennzeichnung* aufzufassen, daß mit diesem Ausdruck genau der Zustand eines Systems S bezeichnet wird, der zusammen mit einem anderen Zustand und S die Satzfunktion $T^*(x, x_1, x_2)$ erfüllt. Dies scheint zumindest dann plausibel, wenn es genau zwei Zustände gibt, die zusammen mit S diese Satzfunktion erfüllen. Denn in diesem Fall könnte man definieren:

(4.9) Dauerleuchten (in S) = derjenige Zustand x_1 von S, für den gilt: $\mathrm{V}x_2\mathrm{T}^*(\mathrm{S}, x_1, x_2)$.

Und

(4.10) Helligkeitsanzeige (in S) = derjenige Zustand x_2 von S, für den gilt: $\mathrm{V}x_1\mathrm{T}^*(\mathrm{S}, x_1, x_2)$.

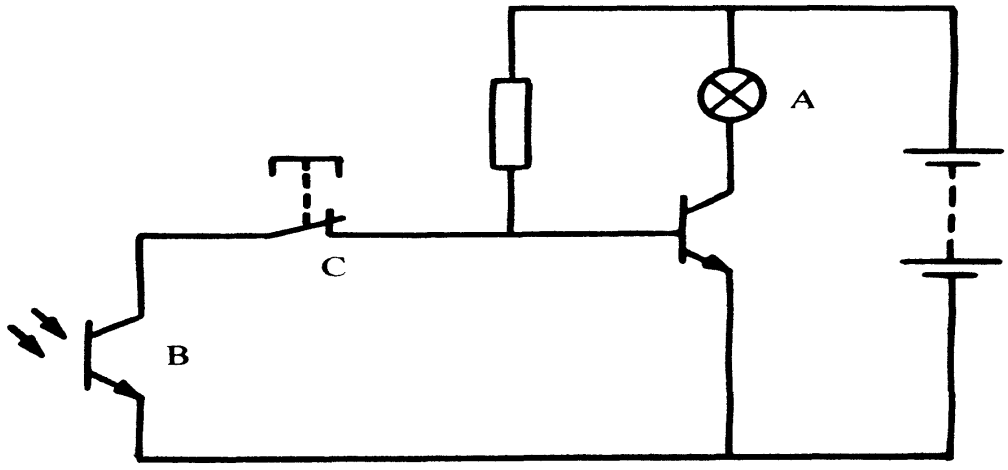
Aus diesen Definitionen folgt aber nach den Gesetzen der Kennzeichnungslogik sofort z.B. ("$\mathrm{V}x!$" steht im folgenden für "Es gibt genau ein x"):

(4.11) Z = Dauerleuchten (in S1*) genau dann, wenn gilt:
$\mathrm{V}x_1!\mathrm{V}x_2\mathrm{T}^*(\mathrm{S1}^*, x_1, x_2)$ und $\mathrm{V}x_2\mathrm{T}^*(\mathrm{S1}^*, \mathrm{Z}, x_2)$.

Falls P1 und P2 die einzigen beiden Zustände sind, die zusammen mit S1* die Satzfunktion $T^*(S1^*, x_1, x_2)$ erfüllen, wird die rechte Seite von (4.10) aber von P1 erfüllt. Zumindest in diesem Fall gilt also auch "P1 = Dauerleuchten (in S1*)". Und entsprechend gilt in diesem Fall dann natürlich auch "P2 = Helligkeitsanzeige (in S1*)".[14]

Zum Abschluß dieses Abschnitts soll noch ein dritter Punkt kurz angesprochen werden: die für den Funktionalismus Putnams charak-

teristische These der *Multirealisierbarkeit* funktionaler Zustände. Diese These wurde von Putnam zunächst am Beispiel von Turing-Maschinen (vgl. unten Abschn. 5) gegen die Identitätstheorie von Place und Smart entwickelt. Sein Hauptargument lautete dabei: mentale Zustände können – ebenso wie die logischen Zustände einer Turing Maschine – als funktionale Zustände nicht mit bestimmten Gehirnprozessen identisch sein, da sie in verschiedenen Systemen auf sehr unterschiedliche Weise materiell realisiert sein können. Abgesehen von der Zielrichtung gegen die Identitätstheorie paßt diese These sehr gut zu den am Anfang des Abschnitts vorgetragenen Überlegungen. Denn wenn man funktionale Zustände als funktional definierte Zustände im Sinne des Schemas (4.4) bzw. der Definitionen (4.7) und (4.8) auffaßt, dann ergibt sich die These der Multirealisierbarkeit a fortiori. Jedes beliebige System S, an dem sich zwei Zustände unterscheiden lassen, die zusammen mit S die Aussagefunktion $T^*(x, x_1, x_2)$ erfüllen, ist der Definition (4.7) zufolge im Zustand "Dauerleuchten", wenn es im ersten dieser beiden Zustände ist – dieser Zustand mag physikalisch oder materiell charakterisiert sein wie er will. Und Entsprechendes gilt auch für funktional definierte Systemtypen. Zwei Systeme S und S′ können physikalisch so verschieden sein wie möglich: wenn sie nur die äußeren Bedingungen erfüllen, die für "umschaltbare Helligkeitsanzeiger" erforderlich sind, und über zwei verschiedene Zustände verfügen, die den genannten Bedingungen genügen, dann fallen beide Systeme unter den Begriff des "umschaltbaren Helligkeitsanzeigers". Vielleicht kann noch einmal ein Beispiel der Verdeutlichung dienen. Denken wir uns dieses Mal ein System S1**, dem der folgende Schaltplan entspricht (der Schaltplan ist aus Gründen der Übersichtlichkeit etwas vereinfacht):



Aus diesem Schaltplan kann man ersehen, daß das System S1**, obwohl es zum Teil aus ganz anderen Bestandteilen aufgebaut ist, im großen und ganzen dasselbe Verhalten zeigt wie das System S1*. Und daraus ergibt sich, daß das Verhalten von S1** ebenfalls mit Hilfe der Theorie T* erklärt werden kann. Darüber hinaus ergibt sich aus dem angegebenen Schaltplan, daß man auch am System S1** zwei physikalische Zustände – nämlich die Zustände "Schalter C ist offen" und "Schalter C ist geschlossen" – unterscheiden kann, die genau den Zuständen Z1 und Z2 der Theorie T* entsprechen. Auch das System S1** ist daher der Definition 4.6 zufolge ein "umschaltbarer Helligkeitsanzeiger".

Das Beispiel des Systems S1** zeigt aber auch, daß aus der Tatsache der Multirealisierbarkeit funktionaler Zustände selbst *nicht* geschlossen werden kann, daß diese Zustände nicht mit physikalischen Zuständen identisch sein können. Denn ebensogut wie man im System S1* den Zustand "Dauerleuchten" mit dem Zustand "Umschalter F ist in Position 1" identifizieren kann, kann man diesen Zustand im System S1** mit dem Zustand "Schalter C ist offen" identifizieren. Die Multirealisierbarkeit funktionaler Zustände spricht zwar gegen eine type-type Identifizierung von mentalen und physikalischen Zuständen; sie widerspricht aber keineswegs einer token-token Identifizierung.

5.

Wenn man die Auffassung, daß mentale Zustände funktional definierte Zustände in dem im letzten Abschnitt geschilderten Sinn sind, *theoretischen Funktionalismus* nennt, dann scheint mir klar zu sein, daß Dennett in diesem Sinne ein theoretischer Funktionalist ist. Denn erstens sagt Dennett ausdrücklich, daß etwas genau dann ein intentionales System sein soll, wenn man sein Verhalten erklären und voraussagen kann, indem man ihm Wünsche und Meinungen zuschreibt (1971, S. 7). Und zweitens sagt auch Dennett über Wünsche und Meinungen nicht mehr, als daß sie bei der Erklärung des Verhaltens intentionaler Systeme eine bestimmte Rolle spielen; auch für Dennett sind Wünsche und Meinungen also nur implizit durch die Theorie charakterisiert, in der diese Zustände zur Erklärung des Verhaltens intentionaler Systeme postuliert werden.[15] Sind in diesem Sinn dann aber nicht auch Fodor und Putnam theoretische Funktionalisten? Wenn man z.B. Fodor (1964) liest, scheint die Antwort auf diese Frage

eindeutig "Ja" lauten zu müssen. Aber bei Putnam liegen die Dinge doch ein wenig anders. Denn es ist ja schon deutlich geworden, daß die Tatsache, daß Putnam in seinem berühmten Aufsatz (1960) seine Theorie am Beispiel von Turing-Maschinen entwickelte, z.B. Dennett – aber nicht nur ihn – zu der Auffassung verleitet hat, der Funktionalismus vertrete die These, alle Wesen mit mentalen Zuständen seien Turing-Maschinen, alle Wesen mit den gleichen mentalen Zuständen hätten die gleiche Maschinentafel und seien insofern in gewisser Weise gleich programmiert, und dergleichen mehr. Explizit vertritt Putnam jedoch nur die These, daß mentale Zustände in gewissem Sinne den gleichen Status haben wie die logischen Zustände von Turing-Maschinen. Klarheit läßt sich daher wohl nur gewinnen, wenn man Putnams Beispiel der Turing-Maschine noch einmal genauer betrachtet.

Bei Turing-Maschinen handelt es sich um sehr einfache Rechenmaschinen. Jede Turing-Maschine arbeitet – in der Normalversion – über einem eindimensionalen, mindestens einseitig unendlichen, in einzelne Felder unterteilten Band, das die Maschine um je ein Feld nach rechts oder links verschieben kann. Außerdem ist jede Turing-Maschine mit einem Schreib-Lese-Kopf ausgestattet, mit dem sie das Symbol erkennen kann, das auf dem Arbeitsfeld (dem Feld, über dem sie sich gerade befindet) steht, und mit dem sie jedes Zeichen (einschl. blank) aus einem vorgegebenen Alphabet auf das Arbeitsfeld drucken kann. Jede Turing-Maschine verfügt also über zwei Bandoperationen "*r*" (Maschine geht ein Feld nach rechts) "1" (Maschine geht ein Feld nach links), je nach Alphabet A über endlich viele Druckoperationen "*dx*" (*x* Element von A oder blank) sowie über eine Stoppoperation "*s*". Die Arbeitsweise jeder einzelnen Turing-Maschine wird durch ihre Maschinentafel bestimmt, aus der hervorgeht, was die Maschine tut, wenn sie auf dem Arbeitsfeld ein bestimmtes Symbol liest – je nachdem, in welchem Zustand sich die Maschine gerade befindet.

Maschinentafeln kann man auf verschiedene Weise darstellen: als Matrizen oder als Folgen von Quadrupeln bzw. Quintupeln. Ich möchte hier die Darstellung in der Form von Folgen von Quadrupeln wählen, weil diese Darstellungsart in diesem Zusammenhang am übersichtlichsten ist. Wenn man eine Maschinentafel als Folge von Quadrupeln beschreibt, dann besagt jedes Quadrupel (a, i, o, b) einer solchen Folge, welche Operation o die Maschine ausführt, wenn sie sich im Zustand a befindet und auf dem Arbeitsfeld das Symbol i liest, und in

welchen Zustand *b* die Maschine im Anschluß an diese Operation übergeht. Die folgende Maschinentafel beschreibt in diesem Sinne die Arbeitsweise einer einfachen Turing-Maschine (sie soll im folgenden M1 heißen), die über dem einelementigen Alphabet {"1"} arbeitet und zu jeder in unärer Darstellung gegebenen natürlichen Zahl *n* den Nachfolger $n + 1$ berechnet, d.h. wenn man die Maschine auf dem Feld direkt rechts neben einer in unärer Form dargestellten Zahl ansetzt, bleibt sie direkt rechts neben dem in unärer Form dargestellten Nachfolger dieser Zahl stehen ("*" steht in dieser Maschinentafel für blank):

MT1 (0, *, *d*1, 1)
(0, 1, *r*, 0)
(1, *, *s*, 1)
(1, 1, *r*, 1).

Die Quadrupel einer solchen Maschinentafel werden häufig als Anweisungen an die Maschine verstanden, aber tatsächlich entsprechen sie eher Verhaltensgesetzen, die besagen, was die Maschine bei gegebener Bandinschrift tut, wenn sie sich in dem durch das erste Symbol des Quadrupels bezeichneten Zustand befindet. Dem ersten Quadrupel der angegebenen Maschinentafel entspricht in dieser Weise z.B. das Gesetz:

(5.1) Wenn die Maschine M1 im Zustand 0 ist, dann druckt sie eine "1" und wechselt in den Zustand 1, falls das Arbeitsfeld leer ist.

Und analog entsprechen den übrigen Quadrupeln die Gesetze:

(5.2) Wenn die Maschine M1 im Zustand 0 ist, dann geht sie ein Feld nach rechts und bleibt im Zustand 0, falls auf dem Arbeitsfeld eine "1" steht.

(5.3) Wenn die Maschine M1 im Zustand 1 ist, dann stoppt sie und bleibt im Zustand 1, falls das Arbeitsfeld leer ist.

(5.4) Wenn die Maschine M1 im Zustand 1 ist, dann geht sie ein Feld nach rechts und bleibt im Zustand 1, falls auf dem Arbeitsfeld eine "1" steht.

Wenn man die Konjunktion dieser vier Gesetze T1 nennt, dann wird die Arbeitsweise der Maschine M1 durch diese Theorie genauso

beschrieben wie durch die zuvor angegebene Maschinentafel. Man kann sogar sagen, daß diese Maschinentafel letztlich nichts anderes ist als eine kondensierte Darstellung dieser Theorie.

Für die Argumentation Putnams scheint mir der zentrale Punkt nun zu sein, daß die logischen Zustände 0 und 1 der Maschine M1 – ähnlich wie oben die Zustände Z1 und Z2 des Systems S1* – nur implizit durch die Theorie T1 charakterisiert sind, d.h. durch das, was diese Maschine bei gegebenem input (Bandinschrift) tut, wenn sie sich in einem dieser Zustände befindet. Aus dieser Tatsache ergibt sich jedenfalls die – wie ich im vorigen Abschnitt schon ausgeführt habe – für den Funktionalismus charakteristische These der Multirealisierbarkeit der logischen Zustände einer Turing-Maschine, d.h. die These, daß die logischen Zustände einer Turing-Maschine – und damit auch diese Maschine selbst – auf die unterschiedlichste Weise physisch realisiert werden können. Wenn nämlich die logischen Zustände der Maschine M1 nur durch die in der Theorie T1 zusammengefaßten Gesetze (5.1)–(5.4) bestimmt sind, dann bedeutet das eben auch, daß jedes physische System D als eine solche Maschine aufgefaßt werden kann, wenn dieses System über einem eindimensionalen, in einzelne Felder unterteilten Band arbeitet, wenn es die für Turing-Maschinen erforderlichen Lese- und Druckvorrichtungen besitzt und wenn sich an ihm zwei (physische) Zustände D0 und D1 unterscheiden lassen, die die für die logischen Zustände 0 und 1 der Maschine M1 charakteristischen funktionalen Eigenschaften haben, d.h. die zusammen mit D die Satzfunktion $T1(x, x_1, x_2)$ erfüllen.

Wenn man Maschinen der Art M1 als Nachfolger-Maschinen bezeichnen will, dann kann man diesen Begriff daher ebenso mit Bezugnahme auf die Theorie T1 definieren, wie im letzten Abschnitt der Begriff des umschaltbaren Helligkeitsanzeigers unter Bezugnahme auf die Theorie T* definiert wurde. Denn nach den vorangegangenen Überlegungen gilt offenbar:

(5.5) x ist eine Nachfolger-Maschine genau dann, wenn gilt:
$$V x_1 V x_2 T1(x, x_1, x_2).$$

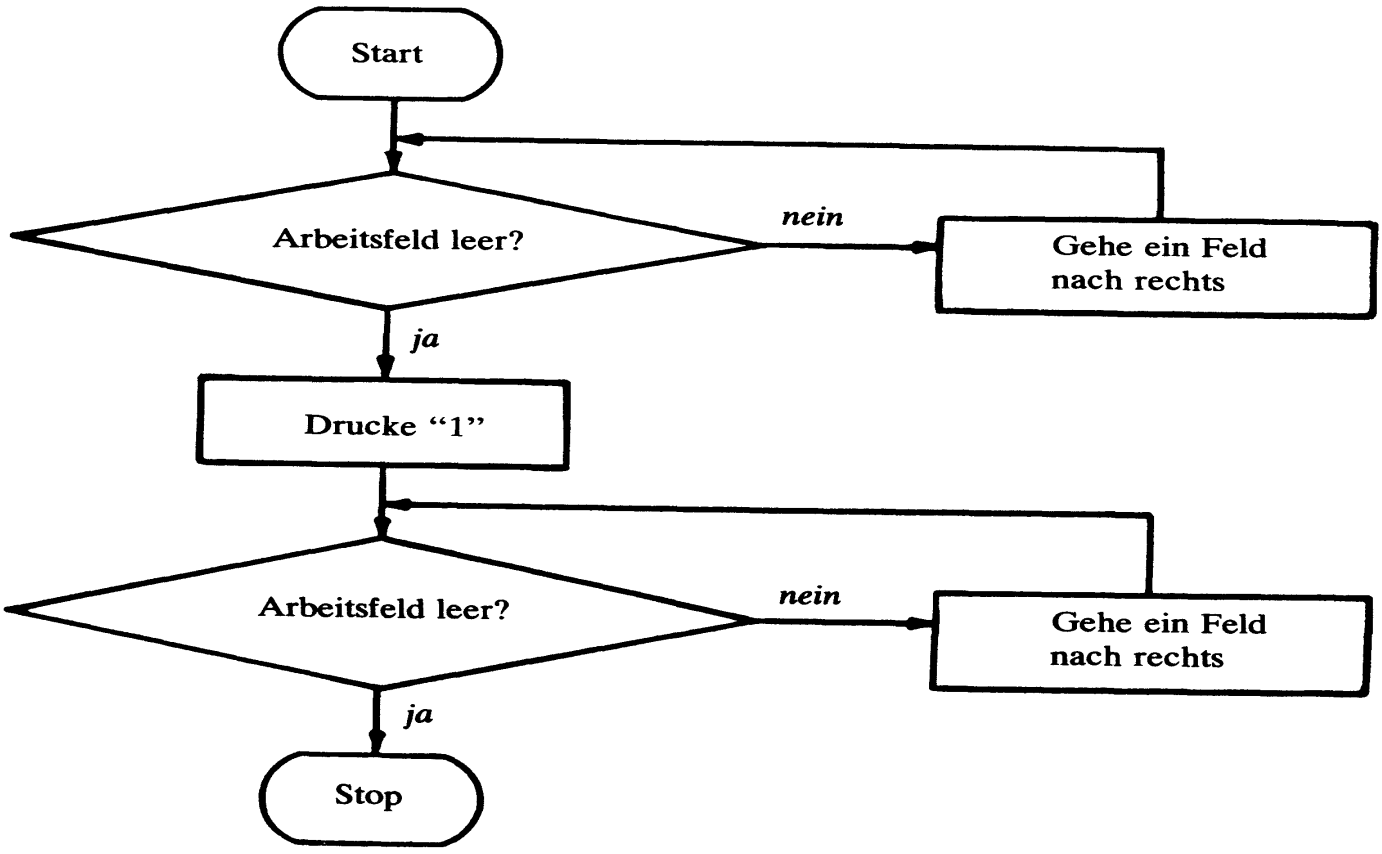
Und entprechend gilt z.B. auch:

(5.6) x ist im Nachfolger-Maschinen-Zustand 0 genau dann, wenn gilt:
$$V x_1 V x_2 (T1(x, x_1, x_2) \text{ und } x \text{ ist in } x_1).$$

Ist also Putnam, wenn er behauptet, mentale Zustände hätten ihrer Natur nach den gleichen Status wie die logischen Zustände einer Turing-Maschine, nicht auch ein theoretischer Funktionalist?

Auf den ersten Blick sieht es zunächst so aus. Doch ganz so einfach ist die Sache nicht. Offenbar kann man die Erklärung des Verhaltens einer Turing-Maschine durch eine Maschinentafel nämlich auch als eine Programm-Erklärung im oben erläuterten Sinn verstehen. Denn der Maschinentafel MT1 z.B. entspricht in ganz natürlicher Weise ein Programm bzw. ein Flußdiagramm. Wenn man dieses Diagramm explizit angeben will, muß man nur berücksichtigen, daß in Maschinentafeln für Turing-Maschinen in der Regel Test- und Operationsanweisungen zusammengefaßt sind. Nach einer Trennung dieser Bestandteile ergibt sich für die Maschinentafel MT1 das folgende Flußdiagramm:

FT1



Für welche Interpretation soll man sich nun entscheiden? Offenbar ist die entscheidende Frage, ob die mit der Maschinentafel MT1 gegebene Theorie T1 eine interne oder eine externe Verhaltenstheorie der entsprechenden Maschine darstellt. Und wenn man von dieser Frage ausgeht, dann scheint Putnam eher an Programm-Erklärungen als an TF-Erklärungen von Turing-Maschinen interessiert zu sein. Denn auf den ersten Blick scheint die Theorie T1 – anders als die Theorie T* – *keine externe* Theorie zur Erklärung des Verhaltens eines gegebenen Systems S zu sein. Zumindest drängt sich diese Antwort auf, wenn man einmal nur die eingegebene Zahl n und das Ergebnis der "Rechnung" der Turing-Maschine betrachtet. Denn wer das Verhalten einer Nachfolger-Maschine sozusagen nur von außen betrachtet, der sieht nur, daß die Maschine rechts neben einer in unärer Form dargestellten natürlichen Zahl n angesetzt nach einer gewissen Zeit rechts neben dem in unärer Form dargestellten Nachfolger $n + 1$ dieser Zahl stehen bleibt. Bei bloß externer Betrachtungsweise reicht also die Zuschreibung einer einfachen Disposition zur Erklärung des Verhaltens dieser Maschine aus. Und wer eine in der eben angegebenen Weise definierte Nachfolger-Maschine nur von außen betrachtet, der kann diese Maschine auch nicht von einer (anderen) Turing-Maschine unterscheiden, die nach der folgenden Maschinentafel arbeitet:

MT2 (0, *, 1, 0)
(0, 1, *l*, 1)
(1, *, *d*1, 2)
(1, 1, *l*, 1)
(2, *, *s*, 2)
(2, 1, *r*, 2).

Denn diese Maschine bleibt, wenn man sie direkt rechts neben einer in unärer Form dargestellten natürlichen Zahl beginnen läßt, ebenfalls nach kurzer Zeit direkt rechts neben dem in unärer Form dargestellten Nachfolger dieser Zahl stehen; sie berechnet also – wenn man so will – ebenfalls die Nachfolgerfunktion. Und ihr Verhalten läßt sich extern durch dieselbe Disposition erklären, mit der man auch das Verhalten der Nachfolger-Maschine M1 erklären kann. Ist also diese Maschine nicht ebenso gut eine Nachfolger-Maschine wie die, deren Arbeitsweise durch die Maschinentafel MT1 bzw. durch die Theorie T1 bestimmt wird?

Nun, das ist natürlich eine Frage der Entscheidung: man kann den Begriff "Nachfolger-Maschine" für Maschinen der Art M1 reservieren,

und man kann auch sagen, daß alle Turing-Maschinen, die die Nachfolgerfunktion berechnen, "Nachfolger-Maschinen" heißen sollen (möglichweise sogar nicht einmal nur Turing-Maschinen). Aber ich denke doch, daß an dieser Frage klar wird, wie die Kritik Dennetts am Funktionalismus insbesondere der Putnamschen Prägung gemeint ist. Dennett ist, so kann man sagen, ein *externer Funktionalist*. Denn seine Theorie scheint mir darauf hinauszulaufen, *daß einem System dann und nur dann bestimmte theoretisch-funktionale Zustände zugeschrieben werden dürfen, wenn die Annahme der Existenz dieser Zustände zur externen Erklärung des Verhaltens dieses Systems sinnvoll bzw. sogar erforderlich ist*. Jedenfalls scheint er mir im Hinblick auf mentale Zustände diese Auffassung zu vertreten. Dennett zufolge sind mentale Zustände theoretisch-funktionale Zustände, die implizit durch eine Theorie charakterisiert sind, *die von einem externen Standpunkt aus das Verhalten entsprechender Systeme optimal erklärt*.

Demgegenüber scheint Putnam eher ein *interner Funktionalist* zu sein. Auch er ist natürlich ein Funktionalist; auch und gerade für ihn sind mentale Zustände funktional definierte Zustände, d.h. Zustände, die nur implizit durch eine Theorie über ihre kausale Rolle charakterisiert sind und die aus diesem Grunde auf die unterschiedlichste Weise physikalisch realisiert werden können. Doch diese Theorie scheint bei Putnam eben doch auf einem bestimmten Wissen über die *innere* funktionale Organisation dieser Systeme zu beruhen. Putnam scheint also, wie gesagt, eher an Programm- als an TF-Erklärungen interessiert zu sein. Und insofern scheint Putnams Version des Funktionalismus tatsächlich zu implizieren, daß gleiche mentale Zustände in gewissem Sinne eine gleiche Maschinentafel oder eine gleiche Programmierung bei den Systemen, um die es geht, voraussetzen. Meiner Meinung nach ist aber nicht ganz klar, ob man mit dieser Interpretation Putnam wirklich gerecht wird. Denn wenn man z.B. bei den Turing-Maschinen, deren Arbeitsweise durch die Maschinentafel MT1 bestimmt wird, nicht nur auf die eingegebene Zahl und das Ergebnis sieht, sondern auch darauf, wie diese Maschinen das Band bewegen und unter welchen Umständen sie welche Zeichen drucken, dann zeigen diese Maschinen doch ein ganz anderes Verhalten als jene Turing-Maschinen, deren Arbeitsweise durch die Maschinentafel MT2 bestimmt wird. Und zur Erklärung dieses Verhaltens benötigt man *auch vom externen Standpunkt aus* tatsächlich eine Theorie wie die

Theorie T1. Mit anderen Worten: Meiner Meinung nach ging es Putnam mit seinem Beispiel der Turing-Maschine zunächst darum, *die Idee des Funktionalismus im allgemeinen* zu erläutern. Vielleicht ist ihm, als er diese Idee zum ersten Mal entwickelte, nicht einmal der Gedanke gekommen, daß man möglicherweise zwischen einem internen und einem externen Funktionalismus unterscheiden könne. Auf jeden Fall kann man aber aus der Tatsache, daß Putnam zunächst Turing-Maschinen als Beispiele für seine Theorie gewählt hat, sicher nicht schließen, daß er bewußt einen *internen* Funktionalismus vertreten wollte.

6.

Zusammenfassend läßt sich meiner Meinung nach folgendes sagen: Dennetts eigene Theorie läßt sich durchaus als eine Art von Funktionalismus kennzeichnen. Denn der Begriff des intentionalen Systems ist für ihn offenkundig ein funktional definierter Begriff im Sinne des Abschnitts 4. Wenn der Intentionalismus Dennetts aber eine Art von Funktionalismus ist, dann eine, bei der, wie Dennett selbst immer wieder betont, keinerlei Annahmen über die innere Struktur der Systeme gemacht werden, die Dennett als intentionale Systeme auszeichnen möchte – sei es nun die physikalische Struktur oder die funktionale Organisation in der Form von Schaltplänen, Programmen oder Maschinentafeln. Dennett ist also, wie schon gesagt, ein *externer* Funktionalist.

Dieser externe Charakter der Dennettschen Theorie zeigt sich ganz deutlich auch an einem bisher noch nicht behandelten Aspekt des Dennettschen Ansatzes: seinem erklärten Instrumentalismus. Dieser Instrumentalismus wird besonders dort deutlich, wo Dennett betont, daß es, wenn wir ein System als intentionales System charakterisieren, nicht darauf ankommt, daß dieses System wirklich Wünsche und Überzeugungen *hat*, sondern nur darauf, daß wir das Verhalten dieses Systems durch die Zuschreibung von Wünschen und Überzeugungen *erklären* und *voraussagen* können. Genauer läßt sich das, was hier mit instrumentalistisch gemeint sein soll, vielleicht wieder anhand des Systems S1* erläutern. Im Abschnitt 4. hatte ich Systeme wie das System S1* als umschaltbare Helligkeitsanzeiger (im folgenen kurz UHA) bezeichnet, wobei dieser Begriff in der folgenden Weise funk-

tional definiert sein sollte:

(4.6) x ist ein UHA genau dann, wenn gilt:
$$\forall x_1 \forall x_2 T^*(x, x_1, x_2)$$

Wenn man für ein bestimmtes System S feststellen will, ob es ein UHA-System ist, muß man daher prüfen, ob für dieses System die Aussage wahr ist

(6.1) $$\forall x_1 \forall x_2 T^*(S, x_1, x_2).$$

Und diese Aussage ist genau dann wahr, wenn es zwei Zustände Z1 und Z2 des Systems S gibt, die zusammen mit S die Aussagefunktion $T^*(x, x_1, x_2)$ erfüllen. Das System S1* ist daher ein UHA-System, weil es in diesem System die beiden Zustände "Umschalter F ist in Position 1" und "Umschalter F ist in Position 2" gibt, die genau dies leisten.

Insoweit scheint die Sache noch unproblematisch. Etwas komplizierter wird es jedoch, wenn wir nicht von der Frage ausgehen, wann Aussagen wie die Aussage (6.1) *wahr sind*, sondern vielmehr die Frage stellen, wann wir Aussagen dieser Art *für wahr halten dürfen*. Denn auf diese Frage scheinen zwei Antworten möglich zu sein. Die erste Antwort könnte lauten, daß wir Aussagen wie die Aussage (6.1) dann und nur dann für wahr halten dürfen, wenn die Zustände, deren Existenz in diesen Aussagen behauptet wird, *explizit angegeben werden können*. Diese Antwort scheint jedoch sehr restriktiv zu sein. Und deshalb könnte eine zweite etwas liberalere Antwort lauten: man darf Aussagen wie die Aussage (6.1) auch schon dann für wahr halten, wenn sie als Theorien über das Verhalten von S *empirisch gut bestätigt sind*. Wenn man diese zweite Antwort für akzeptabel hält, dann kann man sich aber selbst dann eine begründete Meinung darüber bilden, ob z.B. das System S ein UHA-System ist oder nicht, wenn man über die wirkliche innere Struktur dieses Systems nicht das Geringste weiß. Denn in diesem Fall reicht die Tatsache, daß (6.1) eine gut bestätigte Theorie über das Verhalten von S ist, völlig aus, um dieses System als *ein UHA-System klassifizieren zu können*.

Aus verschiedenen Äußerungen Dennetts ergibt sich, daß Dennett eindeutig als Anhänger dieser zweiten Möglichkeit angesehen werden *muß. Denn seine Hauptthese ist, daß ein Objekt genau dann ein* intentionales System ist, wenn man das Verhalten dieses Systems erfolgreich erklären und voraussagen kann, indem man ihm bestimmte Ziele und Meinungen zuschreibt. Insbesondere erfolgreiche Voraus-

sagen des Verhaltens eines Objekts liefern aber umgekehrt eine starke Bestätigung für die Annahme der dem Objekt unterstellten Ziele und Meinungen. Dennetts These kann man deshalb auch so formulieren: ein Objekt ist genau dann ein intentionales System, wenn eine Theorie des Verhaltens dieses Objektes, in der ihm bestimmte Ziele und Meinungen zugeschrieben werden, empirisch gut bestätigt ist.

An manchen Stellen geht Dennett jedoch sogar noch einen Schritt weiter, wenn er behauptet, die wirkliche innere – physikalische oder funktionale – Struktur eines Systems habe überhaupt nichts damit zu tun, ob man dieses System als intentionales System klassifizieren könne oder nicht. Auf das zuvor diskutierte Beispiel angewandt würde diese Auffassung bedeuten, daß es nicht einmal nötig ist, daß es überhaupt identifizierbare physikalische oder strukturelle Zustände gibt, die zusammen mit S die Aussagefunktion $T^*(x, x_1, x_2)$ erfüllen, um das System S als ein UHA-System klassifizieren zu können, wenn nur das Verhalten von S mit Hilfe der Aussage (6.1) erfolgreich erklärt und vorausgesagt werden kann. Die Wahrheit von (6.1) scheint hier mit der guten Bestätigung dieser Aussage zusammenzufallen, wenn man in diesem Zusammenhang überhaupt noch von Wahrheit reden will. Die wirkliche innere Struktur von S jedenfalls hat dieser Auffassung zufolge mit der Wahrheit der Aussage (6.1) nichts mehr zu tun. Darin liegt der Grund dafür, daß ich diese Auffassung "instrumentalistisch" nennen und daß ich dementsprechend auch einen Funktionalismus als "instrumentalistisch" bezeichnen möchte, demzufolge ein System S schon dann bestimmte funktionale Zustände besitzt, wenn nur die Theorie T, in der diesem System diese Zustände zugeschrieben werden, erfolgreiche Erklärungen und Prognosen ermöglicht. Umgekehrt wäre ein "realistischer" Funktionalismus dadurch charakterisiert, daß ihm zufolge ein System S erst dann bestimmte funktionale Zustände besitzt, wenn es tatsächlich innere – physikalische oder strukturelle – Zustände dieses Systems gibt, die zusammen mit S die zu T gehörende Aussagefunktion erfüllen.

Es ist schwer zu sagen, ob man vielleicht in diesem Zusammenhang einen Gegensatz zwischen Dennett und Putnam konstruieren kann. Denn Putnam äußert sich zu diesen Fragen nicht explizit. Für ihn steht eindeutig das Problem der Multirealisierbarkeit funktionaler Zustände im Vordergrund. Dennett jedoch kann, wie mir scheint, sicher als externer instrumentalistischer Funktionalist angesehen werden. Denn einerseits ist für ihn der Begriff des intentionalen Systems offenbar ein

funktional definierter Begriff; andererseits bekräftigt Dennett aber auch immer wieder, daß seiner Meinung nach die Charakterisierung eines Systems als intentionales System nichts mit der wirklichen inneren Organisation dieses Systems zu tun hat.

ANMERKUNGEN

[1] Vgl. 1978a, S. xivff. – Genauer kennzeichnet Dennett seine Theorie (mit einigen zusätzlichen Qualifikationen) als "type intentionalism", wobei er diesen Ausdruck folgendermaßen erläutert: ". . . every mental event is some functional, physical event or other, and the types are captured not by any reductionist language, but by a regimentation of the very terms we *ordinarily* use – we explain *what beliefs are* by systematizing the notion of a believing-system, for instance" (1978a, S. xix).

[2] Vgl. zu dieser Passage auch (1977) und (1978b).

[3] Der Ausdruck "System" soll hier ganz einfach beliebige Entitäten bezeichnen, denen man in irgendeinem Sinn *Verhalten* zuschreiben kann.

[4] Die Termini "intern" und "extern" sind in diesem Zusammenhang sicher nicht ganz glücklich; denn es liegt zunächst sicher nahe zu vermuten, daß interne Verhaltenserklärungen solche Erklärungen sind, die ausschließlich oder überwiegend auf interne Faktoren des Systems Bezug nehmen, und externe Erklärungen dementsprechend solche Erklärungen, in denen dem System externe Faktoren die Hauptrolle spielen. Obwohl dies nicht gemeint ist, habe ich die Ausdrücke "intern" und "extern" in der oben angegebenen Bedeutung hier ihrer Kürze und Prägnanz wegen doch beibehalten.

[5] Der Ausdruck "Schaltplan" ist hier in einem sehr weiten Sinne gemeint, so daß er nicht nur elektrische und elektronische Schaltpläne im eigentlichen Sinn, sondern alle Pläne umfaßt, in denen etwas über funktionale Bestandteile eines Systems und ihre Beziehungen ausgesagt wird. Für solche Pläne gibt es meines Wissens keinen eigenen Namen. Elektrische und elektronische Schaltpläne scheinen mir die einschlägigsten Beispiele für Schaltpläne in diesem weiten Sinne zu sein.

[6] Wieviel Licht genau auf den Widerstand B treffen muß, damit das Relais D den Schalter E öffnet, das ergibt sich aus den genauen elektrischen Daten der betreffenden Bauteile.

[7] Im Prinzip ist sicher klar, wie eine physikalische Erklärung des Verhaltens von S1 aussehen würde. Es ist jedoch nicht uninteressant, einmal zu versuchen, eine solche Erklärung explizit anzugeben. Denn bei diesem Versuch wird deutlich, daß (rein) physikalische Verhaltenserklärungen schon bei relativ einfachen Systemen außerordentlich kompliziert und aufwendig sein können. (Z.B.: Das System S1 besteht erstens aus einem hohlen Glaskörper von bestimmter Gestalt, der weitgehend luftleer gemacht wurde; in diesem Glaskörper befindet sich ein sehr dünner, z.T. spiralförmig gedrehter Wolframdraht . . . ; von dem Glaskörper führt ein drei cm langer, 0.3 mm dicker Kupferdraht zu einem Bauteil, das folgendermaßen aufgebaut ist: . . . usw.)

[8] Ich habe für dieses Programm eine sehr einfache – fast von selbst verständliche – Version der Programmiersprache BASIC gewählt, die heute sehr verbreitet ist.

[9] Ich beziehe mich bei den folgenden Überlegungen insbesondere auf Lewis (1972) und Block (1980b).

[10] Dieser Ramsey-Satz ist insofern äquivalent zu T, als er den gleichen E-Gehalt wie T hat. Dabei ist der E-Gehalt von T die Menge aller Sätze, die aus T folgen und in denen

keiner der Ausdrücke $t_1, \ldots, t_n$ vorkommt. Für die Erklärung des beobachtbaren Verhaltens von S leistet der Ramsey-Satz (4.1) daher das gleiche wie T.

[11] Ein bemerkenswerter Punkt im Hinblick auf diese Definitionen scheint mir zu sein, daß die funktionalen Zustände "Dauerleuchten" und "Helligkeitsanzeige" nur für die entsprechenden Systeme, also nur für "umschaltbare Helligkeitsanzeiger", definiert sind.

[12] Eigentlich handelt es sich hier nicht wirklich um physikalische, sondern um "Schaltplan-Zustände" des Systems S1*; dieser Unterschied ist jedoch in diesem Zusammenhang nicht besonders wichtig.

[13] Auf den Fall, daß es mehrere Zustandspaare gibt, die die Satzfunktion $T^*(S1^*, x_1, x_2)$ erfüllen, will ich hier nicht eingehen. Es liegt aber nahe, in diesem Fall zu versuchen, disjunktive Zustände zu bilden.

[14] In (1972) hat Lewis versucht, diese Gleichungen ohne Rückgriff auf Definitionen wie (4.9) oder (4.10) zu beweisen.

[15] Ich kenne leider keine Stelle, an der Dennett diese Auffassung explizit formuliert. Aber aus dem Kontext aller seiner Arbeiten scheint mir doch klar zu sein, daß er genau auf eine solche Charakterisierung mentaler Zustände hinaus will.

BIBLIOGRAPHIE

Block, N. (ed.): 1980a, *Readings in Philosophy of Psychology*, Vol. 1, Cambridge, Mass.

Block, N. (ed.): 1980b, 'What is Functionalism', in Block (1980a), S. 171–184.

Dennett, D.: 1971, 'Intentional Systems', *JPh* **68**, 87–106 (wiederabgedruckt in (1978a), S. 3–22; zitiert nach dem Wiederabdruck).

Dennett, D.: 1973, 'Mechanism and Responsibility', in T. Honderich (ed.), *Essays on Freedom of Action*, London (wiederabgedruckt in (1978a), S. 233–255; zitiert nach dem Wiederabdruck).

Dennett, D.: 1976, 'Conditions of Personhood', in A. Rorty (ed.), *The Identities of Persons*, Berkeley/Los Angeles/London (wiederabgedruckt in (1978a), S. 267–285).

Dennett, D.: 1977, 'A Cure for the Common Code?', *Mind* **86** (unter dem Titel 'Critical Notice: The Language of Thought by Jerry Fodor'; wiederabgedruckt in (1978a), S. 90–108).

Dennett, D.: 1978a, *Brainstorms*, Montgomery, Verm.

Dennett, D.: 1978b, 'Reply to Arbib and Gunderson', in (1978a), S. 23–38.

Fodor, J.: 1965, 'Explanations in Psychology', in M. Black (ed.), *Philosophy in America*, Ithaca, NY.

Lewis, D.: 1972, 'Psychophysical and Theoretical Identifications', *AustrJPh* **50**, 249–258 (wiederabgedruckt in Block (1980a), S. 207–215).

Putnam, H.: 1960, 'Minds and Machines', in S. Hook (ed.), *Dimensions of Mind*, New York (wiederabgedruckt in Putnam (1975)).

Putnam, H.: 1975, *Mind, Language and Reality, Philosophical Papers*, Vol. 2, London.

Manuscript received 27 June 1984

Philosophisches Seminar
der Georg-August-Universität
Nikolausberger Weg 9C
3400 Göttingen
F.R.G.