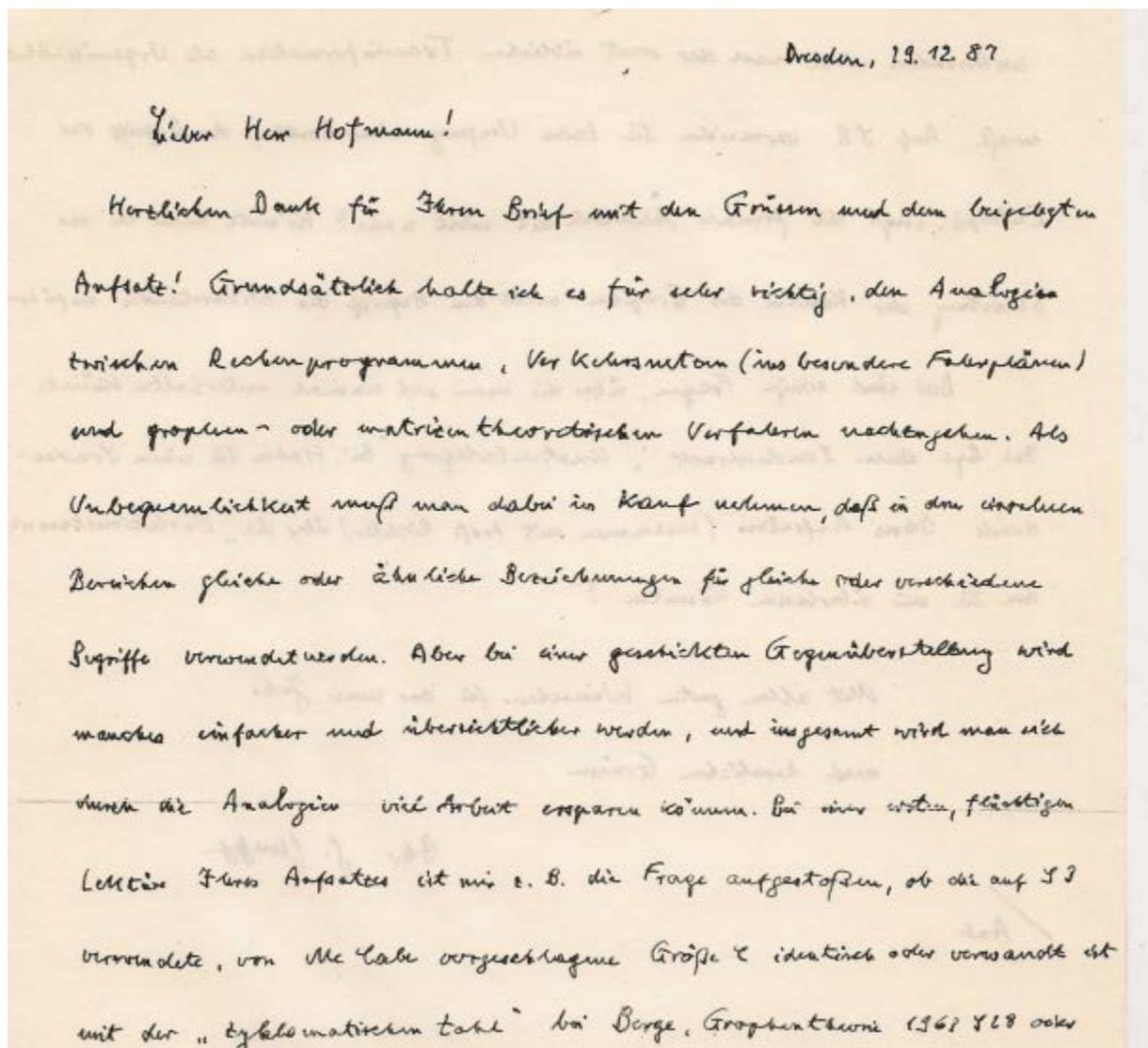


Prof. Potthoff als Berater seiner Schüler

Mit fortschreitender Entwicklung der Fachrichtung Eisenbahnbetrieb wuchs die Anzahl der Schüler, die eigene wissenschaftliche Interessen verfolgten und dies in Veröffentlichung zeigen wollten. Sie suchten dann ihren ehemaligen Lehrer auf und baten ihn um Durchsicht ihrer Manuskripte mit der Bitte um Hinweise und Ratschläge. Bei Prof. Potthoff fanden sie immer offene Ohren und einen konstruktiven Rat.

Dies soll am Beispiel eines Aufsatzes vom damaligen Hochschuldozenten Dr. sc. techn. Ullrich Hofmann (Berlin, dem späteren ordentlichen Professor für Technologie des Eisenbahntransports, zum Thema: „Ansätze für die Bemessung der technologischen Kategorie Komplexität“ (Anlage) berichtet werden.

Prof. Potthoff schrieb unter dem 29.12.1987 an Dr. Hofmann folgenden Brief:



Dresden, 29.12.87

Lieber Herr Hofmann!

Herzlichen Dank für Ihren Brief mit den Grüßen und dem beigefügten Aufsatz! Grundsätzlich halte ich es für sehr wichtig, den Analogietrischen Rechenprogrammen, Verkehrsnetzen (insbesondere Fahrplänen) und graphen- oder matrixtheoretischen Verfahren nachzugehen. Als Unbequemlichkeit muß man dabei in Kauf nehmen, daß in dem unpopulären Bereichen gleiche oder ähnliche Bezeichnungen für gleiche oder verschiedene Begriffe verwendet werden. Aber bei einer geschickten Gegenüberstellung wird manches einfacher und übersichtlicher werden, und insgesamt wird man sich durch die Analogie viel Arbeit ersparen können. Bei einer weiteren, flüchtigen Lektüre Ihres Aufsatzes ist mir z. B. die Frage aufgestoßen, ob die auf S 3 verwendete, von Mc Call vorgeschlagene Größe ξ identisch oder verwandt ist mit der „topologischen Zahl“ bei Berge, Graphentheorie (S 6) 418 oder

mit der „Erschwerungsphase“ bei Knödel, Gröbneralgorithmische Methoden
1969 S. 8. Die Bemerkung der „Erreichbarkeit“ auf S. 9 sieht der Matrizen-
methode ähnlich, die M. Hasse, Optimale Wege verwendet und auf die ich mich
in der Verkehrsströmungslehre Bd. 3 S. 68 beziehe. Im Bild 2 (S. 56) verwenden
Sie den Begriff „Adjazenzmatrix“, der von anderen Verfassern als „Zusammenhangs-
matrix“ bezeichnet wird. Sie werten aber diese Matrix nicht nach ihren Entropien aus,

insbesondere nicht nach der somit üblichen, Transinformation als Organisations-
maß. Auf S. 8 verwenden Sie beim Verlauf einer Routine den Begriff der
Entropie. Liegt die formale Ähnlichkeit nicht nahe? Könnte man bei der
Bewertung der Kanten des Graphen nicht den Begriff des Widerstandes einführen?

Das sind einige Fragen, über die man sich einmal unterhalten könnte.
Ich habe einen Sonderdruck „Knotenbelegung“ bei. Haben Sie einen Sonder-
druck Ihres Aufsatzes (zusammen mit Prof. Richter) über die „Evolutionstheorie“,
den Sie mir überlassen könnten?

Mit allen guten Wünschen für das neue Jahr
und herzlichem Grüßen

Herr J. Jurek.

Aul

Auch der Unterzeichnete wurde um Rat gefragt.

lieber Herr,
 anbei ein Vorschlag, die
 technologische Komplexität
 zu verstehen. Dabei habe ich auf die
 Systemähnlichkeit von Software-
 und Technologieentwicklung zu-
 rückgegriffen.
 Über diese Fragen würde ich gerne mit
 dir demnächst Eörterungen führen.
 Mit den besten Wünschen
 und freundlichen Grüßen
 Hei

Dr. Hofmann erhielt folgende Antwort in Form einiger fachlicher Notizen mit mündlichen Hinweisen.

Komplexität:
 Art und Anzahl der bestehenden Verbindungen zwischen
 den Elementen eines Systems.
 Maß: Komplexitäts- oder Kopplungsgrad

$$g = \frac{k}{k_{\max}}$$

$$k_{\max} = N(N-1)$$

$$N = \text{Anzahl der Elemente}$$
 k - Anzahl der vorhandenen
 Kopplungen
 N - Anzahl der Elemente
 Messung der Komplexität beschränkten Ein- u. Ausgängen:
 Relative Entropie

$$h = \frac{H}{H_{\max}}$$
Komplexität:
 Unterscheidbarkeit der Funktion von Systemelementen
 Maß der Differenzierbarkeit der Elemente ist Komplexitätsgrad.
 → relative Redundanz

$$r = 1 - h$$

$$g = r = 1 - \frac{\sum \hat{t}_p \lg \hat{t}_p}{\lg N}; \quad \hat{t}_p = \frac{t_p}{\sum t_p}$$

$$t_p - \text{benötigte Transformation des Elements } E_p$$

Strukturreffektivität:

- Zentralitätsgrad eines Elements
- Erreichbarkeit

Kennzahl für peripheren Lage des Elements p

$$d_p^* = \frac{\sum_p d_{pq}}{\sum_{p,q} d_{pq}}$$

d_{pq} - kleinste Anzahl
benachbarter Knoten in einem
Netzwerk, über die E_p mit
 E_q verbunden sind

$$0 \leq d_p^* \leq 1$$

→ Element ist weit vom Zentralelement
entfernt.

$$d_p^{**} = 1 - d_p^*$$

Charakteristik für die zentrale Lage
eines Elements E_p
Zentralitätsgrad

$$d_{\max}^{**} - \text{Zentralelement}$$

$$d_p^{***} = d_{\max}^{**} - d_p^{**}$$

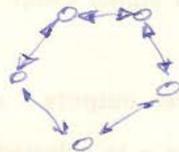
Maß für die Lage von E_p
zum Zentralelement

$$\sigma = \sum_p d_p^{***}$$

Systemzentralitätsgrad

Beispiel: Ring mit $N > 4$

$$\sigma_R = \frac{N-4}{2(N-2)}$$



Gez. Prof. Krampe

Leider kam es nicht zu einem Abschluss dieses Vorhabens, da Kollege Hofmann 1988 direkt an der HfV in Dresden zum Einsatz kam. Die neuen Belastungen durch die vielfältigen Aufgaben am WB Technologie des Eisenbahntransports zwangen ihn offenbar zur Rückstellung des Aufsatzes und damit zum Aufschub einer beabsichtigten Veröffentlichung

Ansätze für die Bemessung der technologischen Kategorie
Komplexität

1. Ausgangsbetrachtungen

Die evolutionstheoretische Betrachtung von Technologien des Status quo und dessen Veränderungen führt zur Aussage, ob bei der jeweiligen Veränderung eine echte Weiterentwicklung hin zu einem höheren Niveau an Komplexität und Organisiertheit vorliegt.

Derartige Untersuchungen sind insbesondere bei der Entwicklung neuer Technologien im Zusammenhang mit computergestützten Prozessen nach dem CIM- bzw. CAI-Konzept zur Sicherung einer hohen Effizienz notwendig. Über erste Gedanken in dieser Richtung sowie zu Ergebnissen der ausgeführten Bewertungen von technisch-technologischen und organisatorischen Prozessen im Verkehrswesen wurde in /1/ berichtet. Dabei wurde nur grob auf die Methodik zur Bemessung der Komplexität dieser Prozesse als ein Maßstab des Entwicklungsniveaus eingegangen.

Nunmehr wird der Versuch unternommen, dazu erste konkrete Vorstellungen aufzuzeigen. Dabei besteht bewußt die Absicht, Erfahrungen anderer Wissenschaftszweige aufzugreifen, das Allgemeingültige herauszustellen und auf die Transporttechnologie zu übertragen bzw. modifiziert anzuwenden. Denn in der gegenwärtigen Phase der Entwicklung der Wissenschaft haben die informellen Prozesse in physikalischen, organischen, technischen und gesellschaftlichen Systemen, einschließlich der Mechanismen der Informationsbildung und -speicherung sowie der Informationsweitergabe und -verarbeitung, innovative Wirkungen auf andere Wissenschaftszweige. Deshalb werden die Erkenntnisse aus der Wertung von Software hinsichtlich ihrer Komplexität gezielt analysiert und auf technologische Prozesse bezogen, da nach der Ähnlichkeitstheorie hinsichtlich des Bewertungs-, Ordnungs- und Beschreibungspostulats hier eine Systemübereinstimmung zu Transporttechnologien hergestellt werden kann.

Zur Lösung der Systemähnlichkeit ist es jedoch erforderlich, den technologischen Ablauf zu algorithmieren bzw. graphentheoretisch aufzuarbeiten. Dabei muß jedoch von Anfang an festgestellt werden, daß es nicht Ziel der Beurteilung der Komplexität sein kann, den einzelnen konkreten technologischen Arbeitsgang in einer Komplexitätsbeurteilung zu erfassen, denn die Vorgehensmethoden und jeweiligen Randbedingungen sind zu unterschiedlich.

Was ist Komplexität?

Vielmehr bieten sich in Anlehnung an Softwarebeurteilungen zur Bildung von Komplexitätsgrößen grundsätzlich zwei Wege an:

- Die Komplexität wird ausgedrückt in der Anzahl Schritte eines abstrakt gehaltenen Bearbeitungsvorganges zur Erreichung des vorgegebenen Ziels (Theorie des Antizipierens);
- indirekt kann auch die Komplexität nach der Fehleranfälligkeit dieses Bearbeitungsvorganges bewertet werden.

In solch einem konkreten Fall muß die Randbedingung des abstrakten Bearbeitungsvorganges so festgelegt sein, daß weitgehend nur der betrachtete relevante Gesichtspunkt zur Geltung kommt, an dem die Bewertung entwickelt wird.

Zum Beispiel wird bei einem Softwaretest, bei dem als Testabdeckungsgrad das Durchlaufen jedes Programmpfades vorgeschrieben ist, eben die Anzahl Pfade die relevante Größe zur Bestimmung der Anzahl der Testdaten verwendet. Diese Anzahl ist jedoch situationsabhängig und eine einfache Auswertung nicht möglich.

Es bietet sich deshalb der Weg an, die Komplexität einer Aufgabe situationsunabhängig zu bestimmen. Zu diesem Vorgehen, das für Analysen von Transporttechnologien hinsichtlich deren Komplexität besonders geeignet erscheint, soll näher eingegangen werden.

Bei der Ermittlung einer rein objektbezogenen Form der Komplexität werden in jüngster Zeit verschiedene Ansätze verwendet. Sie sind aus

- der Graphentheorie,
- der Sprach-/Informationstheorie,
- der Theorie der Rechenkomplexität und
- der Logik

abgeleitet worden.

2. Graphentheoretische Ansätze zur Komplexität

Im Entwicklungsprozeß eines Rechenprogrammes oder eines Algorithmus für einen technologischen Prozeß treten viele Graphen und Sachverhalte, die durch Graphen wiedergegeben werden können, auf. Das ist auch in der Transporttechnologie, z. B. beim Fahrplan der Eisenbahn, der Fall.

Das muß vorher abgeleitet werden

Die Darstellung solcher Abläufe mit Graphen führt zu charakteristischen Abbildungen des Graphens mit seinen Kanten und Knoten (Bild 1).

Der einzelne Knoten ist aus der Sicht der Einbettung in den Graphen durch die Anzahl der ihn betreffenden Kanten definiert.

Betrachtet man nun ein Flußdiagramm oder einen Algorithmus, so ist dies ein Graph, der alle möglichen Verzweigungen und Zusammenführungen des Kontrollflusses darstellt.

Als Bild 2 ist beispielhaft ein FORTRAN Steuerfluß als Graph wiedergegeben. In diesem Graph umfassen die Knoten alle "Steueranweisungen" und die Kanten entsprechen Programmsegmenten, den vom Kontrollfluß her gesehen unteilbaren sequentiellen Einheiten, die auch leer sein können.

Bei der Analyse solcher Graphen ist die reine Anzahlbetrachtung von Knoten und Kanten nur ein grober Maßstab. Aussagekräftiger sind Bewertungen hinsichtlich der strukturellen Eigenschaften. So ist bedeutsam, ob der Graph Mehrfachkanten und Schlingen aufweist und die Form der Abgeschlossenheit des Graphen (Bilder 3 a - 3 f).

Es stellen sich somit die Fragen:

- Ist der Graph geschlossen?
- Gibt es Teilgraphen oder isolierte Punkte?
- Hat der Graph mehrere Ein- bzw. Ausgangsknoten ?

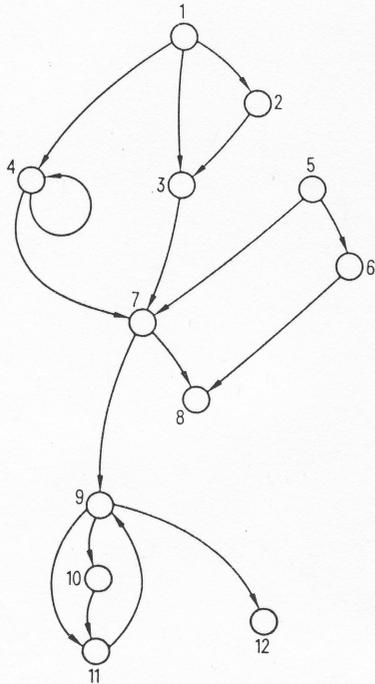
Hat der Graph keine isolierten Teile, so interessieren weitere Strukturmerkmale:

- Ist es ein binärer Baum?
- Ist es überhaupt ein Baum?
- Gibt es Teilgraphen bestimmten Typs?

Graphen dieser Strukturmerkmale zeigt das Bild 4.

Stelle man nun die Komplexitätsfrage in solch einem Bild des Graphen, so geben die genannten Strukturmerkmale Hinweise auf die Diagnosestrategie und die Kantenzüge auf die einzelnen Suchwege. Der Zusammenhang dieser Aspekte ergibt sich dadurch, daß sich eine einfache Struktur eben dadurch auszeichnet, daß - relativ zur Gesamtgröße - nur wenige Kantenzüge möglich sind bzw. die Züge sich uniform zusammensetzen.

Als Zahlenwert kann die von Mc Cabe verwendete Größe C, wie sie in /2/, /3/, /4/ näher behandelt wird, Anwendung finden.



Anzahl Kanten: 17
Anzahl Knoten: 12
Grade: 1:3
2:2
3:3
4:4
5:2
6:2
7:5
8:2
9:5
10:2
11:3
12:1

Bild 1

Graph mit Anzahl Kanten und Knoten

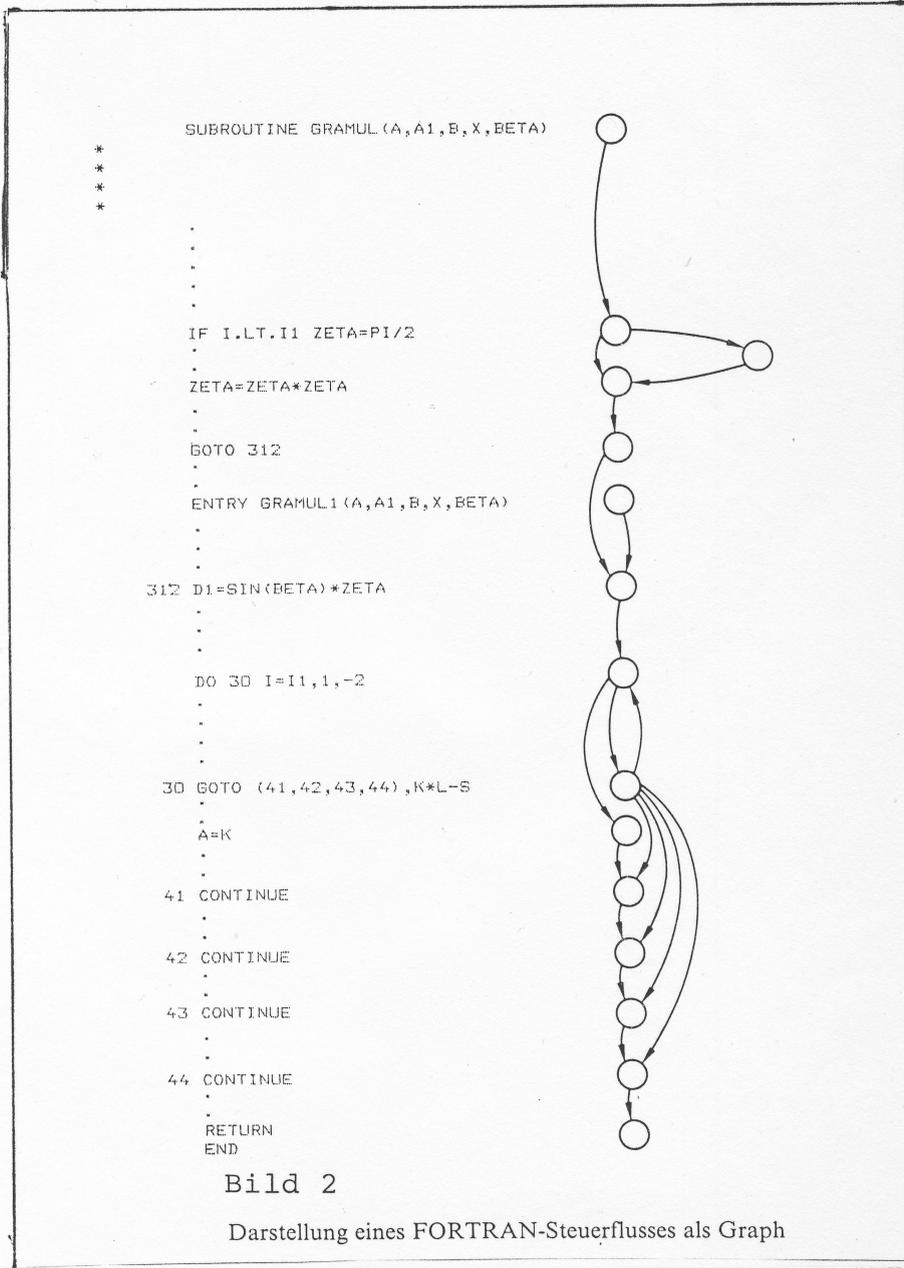


Bild 2

Darstellung eines FORTRAN-Steuerflusses als Graph

hier sollte als Beispiel der Steuergraph zu Bilden ein
Fehlertypus angegeben werden.

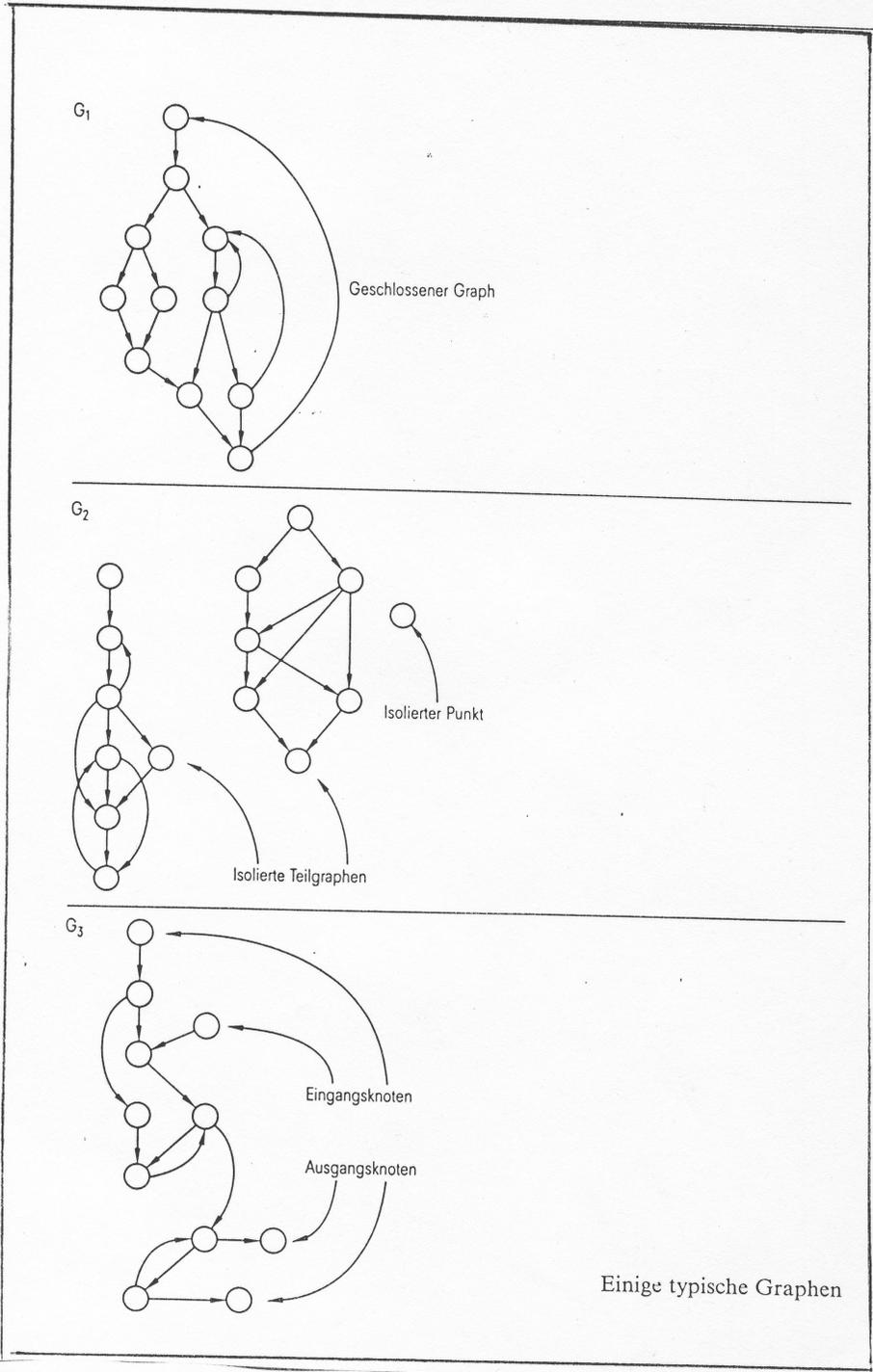
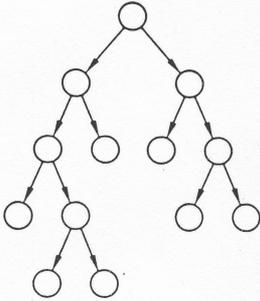


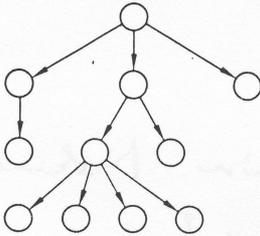
Bild 3 a - c

G₄



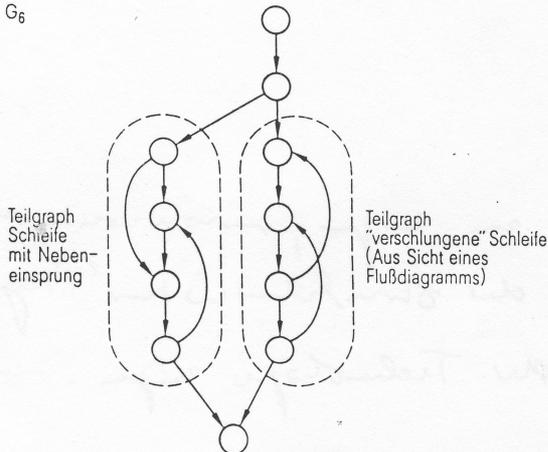
Binärer Baum

G₅



Baum

G₆



(Fortsetzung)

Sie ergibt sich aus der Anzahl der Kantenzüge, die - ohne Überschneidung - minimal notwendig ist, um alle Kantenzüge durch Kombinat bilden zu können. Für einen geschlossenen Graphen kann sie leicht berechnet werden:

$$c = e - k$$

e = kanten (edges)

k = knoten (knots).

Der beispielhaft aufgezeigte Kontrollflußgraph (Bild 5) ohne den sekundären Entry hat die Größe

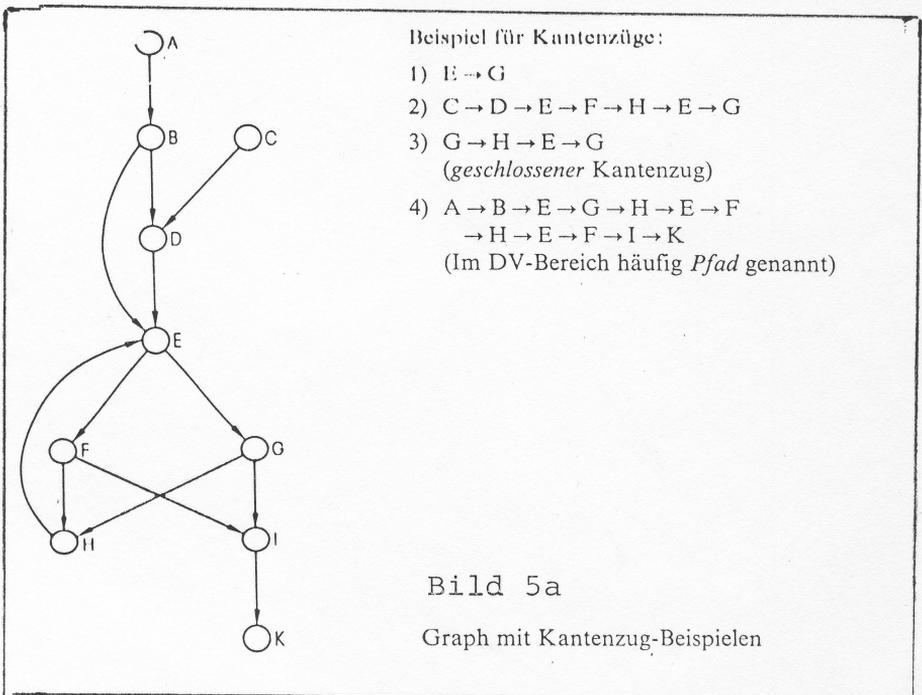
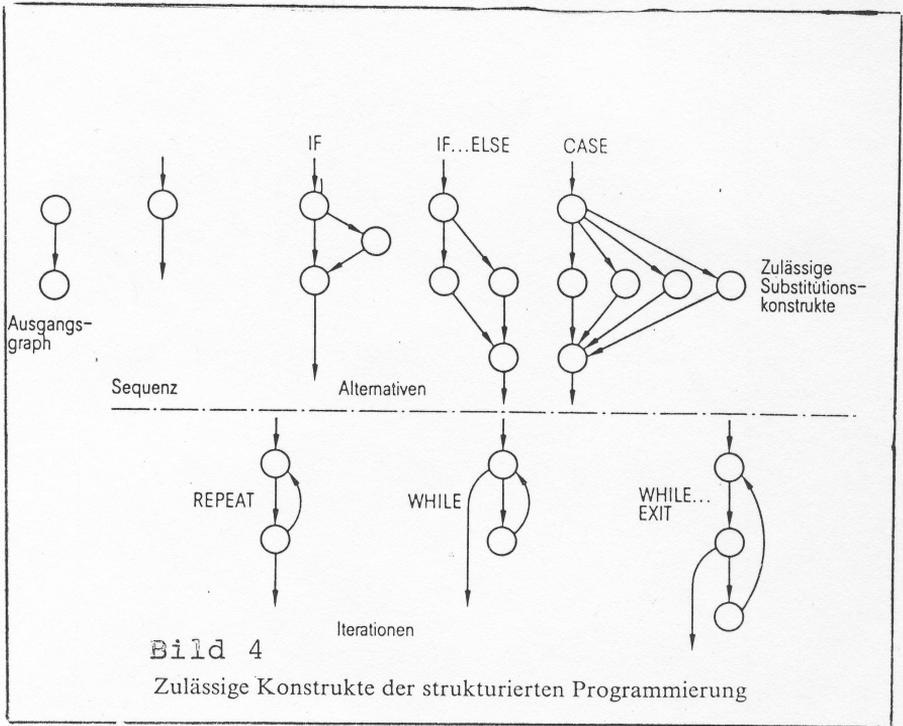
$$C = 12 - 9 + 1 = 4.$$

Dabei ist 1 eine Pseudokante, die den Ausgangsknoten K mit den Einigungsknoten A verbindet und somit den geschlossenen Graphen schafft.

Lb
zusammen?
Diese McCabe Größe bildet eine gute Abschätzung, z. B. für die untere Schranke der Anzahl Testfälle einer Software, die notwendig sind, um die Testbedingung: jede Kante, jede Anweisung zu erreichen, letztlich zu erfüllen.

Aus der Sicht der Bewertung der Komplexität in der Technologie kann mit dieser Größe die Anzahl der möglichen Verfahrenswege zur Erreichung des Zieles im technologischen Prozeß beurteilt werden. Diese Aussage gewinnt insbesondere bei der Entwicklung computergestützter Lösungen in der Produktionssteuerung nach der CIM- oder CAI-Konzeption an Bedeutung. Denn hier geht es darum, mit der neuen Technik eine neue Technologie zu entwickeln, die sich durch stark ausgeprägte Grundmodule auszeichnet und eine Konzentration auf den Teil der flexibel zu steuern- den technologischen Prozesse zum Inhalt hat /1/.

Inwiefern die neue Technologie gegenüber dem Status quo diese Eigenschaft der höheren Komplexität besitzt, kann quantitativ mit der McCabe Größe grob bewertet werden. Betrachtet man nun die McCabe Größe zur logischen Komplexität, insbesondere bei Analysen einer vorgegebenen Struktur, so wirkt sich negativ aus, daß Alternativen und Iterationen im Ablauf (z. B. im Rechenprogramm) gleich behandelt werden. Dies obwohl es kaum vergleichbare logische "Niveaus" sind. So hat eine Routine mit einem CASE mit vielen Labels einen wesentlich höheren McCabe Wert als eine Routine, die aus einer verschränkten zweifachen Iteration besteht. Trotz dieser Schwäche ist die McCabe Größe als eine Kenngröße bei Komplexitätsbewertungen brauchbar.



Aber die Graphenauswertung bietet weitere Bewertungsmöglichkeiten. So ist es möglich, die Pfade und Kantenzüge, die vom Eingang- bis zu einem Ausgangsknoten führen, zu erfassen. Für eine Betrachtung der Komplexität der Abläufe kann über die Pfade jede Graph-Absuchaktion, sei sie analytisch oder über Test, am allgemeinsten erfaßt werden (Bild 6). Hat man nun als Grundmodell einen Zustandsvektor zu Beginn, der bei Durchwandern des Graphen zu einem Ergebnisvektor transformiert wird, so ist die Anzahl der Suchaktionen bei fehlerhaftem Ergebnisvektor und fehlender zusätzlicher Information durch die Pfadanzahl bestimmt. Diese Betrachtungsweise der Komplexität ist für die Qualitätssicherung und -analyse in der Technologie ein neuer Aspekt.

Eine weitere Komplexitätsbetrachtung beleuchtet die Frage der Abhängigkeit innerhalb des Graphen. Von seiten der Graphentheorie wird hier die Erreichbarkeit eines Knotens beschrieben. Es gilt, daß ein Knoten von einem anderen erreichbar ist, wenn ein Kantenzug von diesem zu jenem existiert. Bei einem geschlossenen Graphen sind alle Knoten von allen aus erreichbar; es erreicht jeder sich selbst über einen geschlossenen Kantenzug. Dagegen ist bei isolierten Teilen kein Knoten eines anderen Teils erreichbar, und dies von beiden Seiten her betrachtet.

Zur Bemessung der Erreichbarkeit kann nach Auswertung des Graphs in Form einer Matrixendarstellung auch die Quantifizierung erfolgen.

Die im Beispiel (Bild 7) zu G gehörige Matrix, die deren Kanten beschreibt, ist eine Boolesche Matrix. Wird diese multipliziert

$$M_G^2(n, m) = \sum_{\alpha \in A}^k M_G(n, \alpha) \cdot M_G(\alpha, m),$$

so gilt

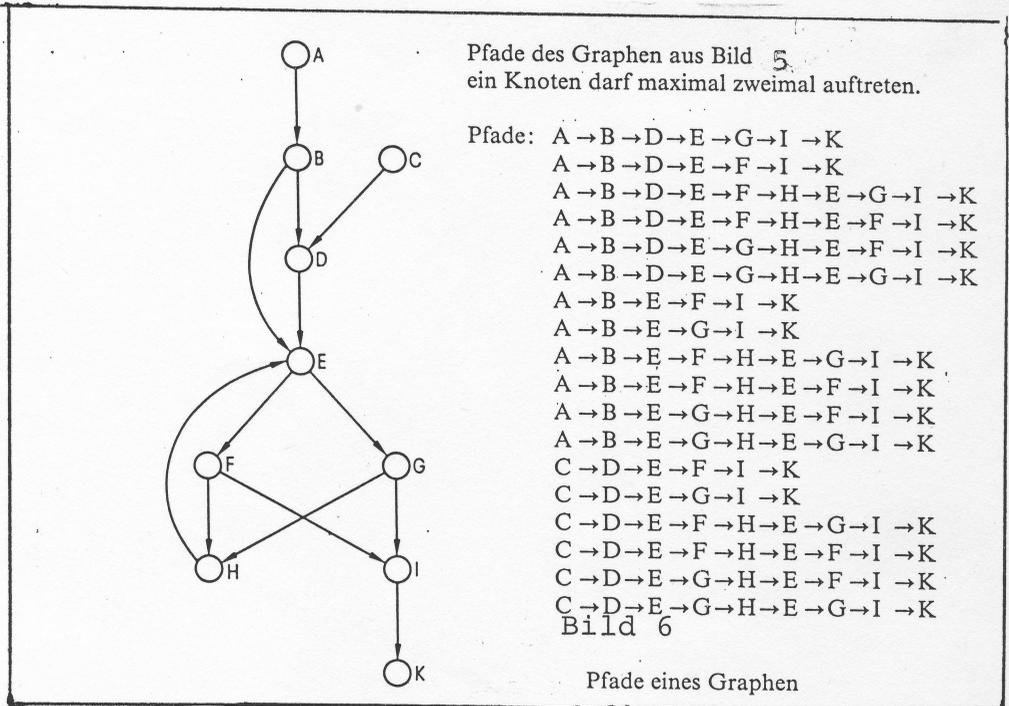
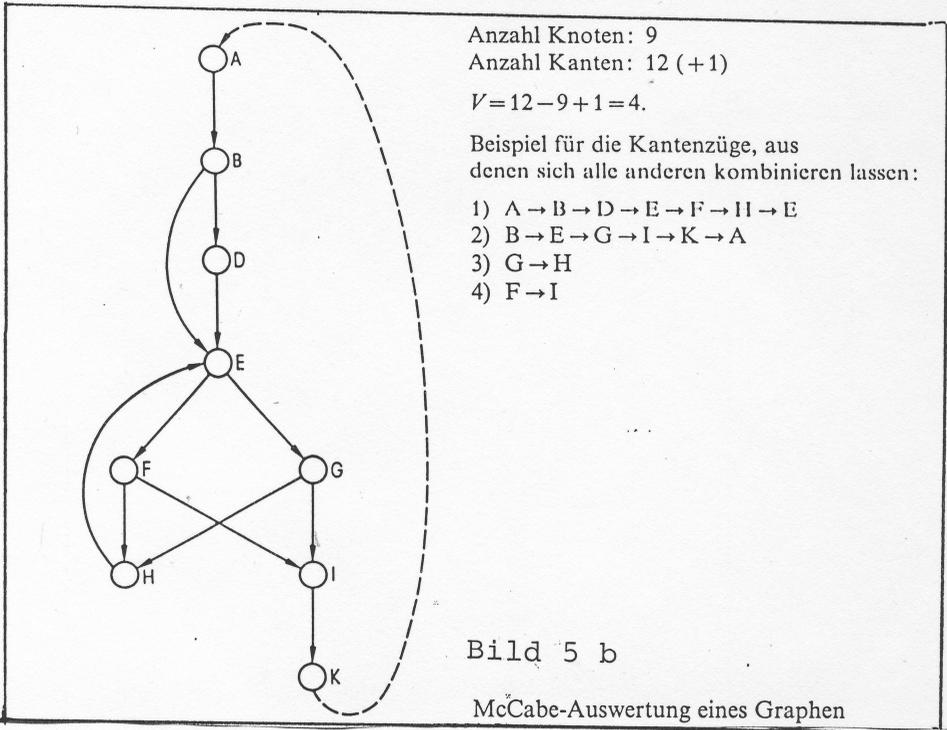
$$M_G^2(n, m) = L \Leftrightarrow p (M_G(n, p) = L \wedge M_G(p, m) = L),$$

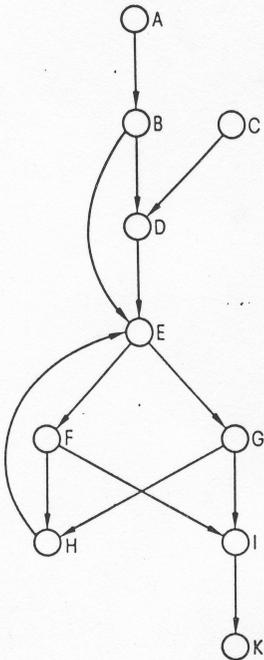
d. h. M_G^2 ist die Matrix, die alle Kantenzüge der Länge 2 darstellt.

Dies trifft auch für höhere Potenzen zu. Geht man nun über zur Operation

$$M_G^n = \sum_{l=1}^n M_G^l,$$

so weist diese Operation einen Fixpunkt auf.





M_G	A	B	C	D	E	F	G	H	I	K
A	0	L	0	0	0	0	0	0	0	0
B	0	0	0	L	L	0	0	0	0	0
C	0	0	0	L	0	0	0	0	0	0
D	0	0	0	0	L	0	0	0	0	0
E	0	0	0	0	0	L	L	0	0	0
F	0	0	0	0	0	0	0	L	L	0
G	0	0	0	0	0	0	0	L	L	0
H	0	0	0	0	L	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	L
K	0	0	0	0	0	0	0	0	0	0

In G gibt es eine Kante
von X nach $Y \Leftrightarrow M_G(X, Y) = L$

Bild 7

Graph mit seiner Adjazenzmatrix

Die Matrix \hat{M} ist die Erreichbarkeitsmatrix. Denn

$$\hat{M}_a(X, Y) = L ; \text{ d. h. es gibt}$$

einen Kantenzug zwischen X und S, S ist von X aus erreichbar. Y?

Bei der Softwareanalyse hat die Frage der Erreichbarkeit bei der Bewertung von "Fehlerausbreitungen" Bedeutung. Es kann die Frage beantwortet werden, wohin ein Fehler überall ausstrahlen bzw. umgekehrt - wenn ein "interner" Fehlerort bekannt ist -, woher der Fehler gekommen sein kann.

Diese Frage stellt sich auch bei der Qualitätsbewertung von technologischen Prozessen. Denn auch hier gilt es bei unterschiedlicher Komplexität der technologischen Abläufe, die Fehlerauswirkung und -ausbreitung analytisch zu bestimmen und somit auch verschiedene Varianten unter diesen Aspekten bewerten zu können.

Zu diesen bisher skizzierten Möglichkeiten der graphentheoretischen Betrachtungsweise der Kategorie Komplexität wurde von reinen Kanten/Knoten-Kriterien ausgegangen. Häufig kann aber auch zu bewerteten Graphen übergegangen werden: jeder Kante ist ein "Gewicht" zugewiesen. So kann z. B. bei Pfaden der Kante die Wahrscheinlichkeit der Auswahl zugewiesen werden, bei Erreichbarkeitsfragen kann ein Bedeutungsgewicht berücksichtigt sein. Ein solches Verfahren im Bereich der Softwarebewertung wurde in /5/ beschrieben und wird hier als "Modularitätsbewertung" bezeichnet. Diese Methodik der Verwendung von bewerteten Graphen hat sicher ein noch weit offenes Feld im speziellen Anwendungsbereich der Qualitätsbewertung von Transporttechnologien. Die Vergleiche zwischen Softwarebewertung und Bewertung von Transporttechnologien nach der Ähnlichkeitstheorie zeigen, daß hier die Übernahme der graphentheoretischen Verfahren z. B. für die Belange der Fahrplanbewertung der Eisenbahn möglich ist. Aou?

3. Komplexität unter dem Aspekt der Elementevariation

Der wieder zum Vergleich herangezogenen Softwareentwicklungsprozeß kann aus der Sicht der Systemanalyse als ein sprachlicher Transformationsprozeß verstanden werden. Eine Aufgabe, die in einer bestimmten Sprache formuliert wurde, soll in eine andere übertragen werden. Es kann auch umgekehrt die Analyse einer formal formulierten Aufgabe als Rückübersetzung verstanden werden.

Bei der Entwicklung von Technologien haben wir im Grunde nach Systemähnlichkeit zum Softwareentwicklungsprozeß.

Die Schwierigkeit und Fehleranfälligkeit dieser Transformationen läßt sich in verschiedener Hinsicht durch die Analyse der sprachlichen Struktur (Elementestruktur im technologischen Ablauf) der jeweiligen Ergebnisse beleuchten. Diese Betrachtungsweise wurde bisher ausschließlich in der Softwareentwicklung auf den Quellcode begrenzt, wobei die Ausgangssituation und das Ergebnis des Desingprozesses unbestimmend gelassen wurden /6/.

Ferner wurde eine ähnliche Betrachtungsweise bei einer ersten Bewertung eines CIM-Konzeptes angewendet /1/. Dabei wird vom einfachsten Ansatz der syntaktischen Analyse der Sprache (technol. Elemente) ausgegangen und eine sogenannte Verwendungsaufzeichnung erstellt /7/. Sie enthält folgende Angaben:

- Welche Schlüsselwörter der Sprache treten wie oft auf?
- Wie viele Parameter, globale Variable, lokale Variable, konstante etc, wurden definiert?
- Wie streut die Verwendung der Variablen, d. h. in wieviel tritt die Variable auf? Wie groß ist der Abstand - gemessen in Anweisungen (techn. Arbeitsgängen) - zwischen dem ersten und letzten Auftreten einer Variablen?

Aus den Angaben über die Häufigkeit von Konstruktionen, wie

- Anteil von Alternativ-, Iterationsbezeichnern an der Zahl der Anweisungen,
- Verhältnis lokaler und globaler Variablen,
- Schachtlungstiefe von Alternativen, Iterationen etc.,

kann man dann auf eine generelle, unbestimmte Schwierigkeit der Aufgabenstellung, als ein weiteres Bewertungskriterium der Kategorie Komplexität, schließen.

Eine konkrete Methodik dazu hat Healstead in Form einer Monographie entwickelt /8/. Seine Ausgangsgrößen sind

W der Wortschatz, die Anzahl der verwendeten sprachlichen (technologischen) Grundeinheiten und

N die Länge des Satzes, d. h. die Länge der Sequenz aus Elementen des Wortschatzes, die die Routine bilden.

Der Quellcode ist hier nicht als Übermittlungscode an die Maschine, sondern als Ausdrucksmedium des Programmierers zu verstehen.

Unter Bezug auf die Informationstheorie kann man zu einer normalisierten Größe der betrachteten Routine kommen, die als Umfang der Routine (V) bei Softwareanalysen bezeichnet wird

$$V = N \cdot \text{lb} (\eta).$$

Dies ist die Länge des Satzes bei einer optimalen Binärcodierung der Wörter, wobei - aus Vereinfachungsgründen - oft eine Gleichverteilung der Wörter vorausgesetzt ist.

Nach der Halstead-Methodik werden die auftretenden Wörter weiter in Operatoren und Operanden (die Operatoren, die von der Sprache vorgegebenen Wörter, die Operanden, die vom Programm frei geprägten Wörter - das sind in erster Linie alle Variable -) unterteilt. Dies ist prinzipiell auch bei einem algorithmierten technologischen Ablauf möglich. Es sind also

$$\begin{aligned} \eta_1 & \text{ Anzahl der Operatoren und} \\ \eta_2 & \text{ Anzahl der Operanden} \\ \eta & = \eta_1 + \eta_2 \end{aligned}$$

Als System betrachtet muß die Routine mit Parametern versorgt werden, die von Ein- und Ausgabe transportiert werden müssen. Eine minimale Formulierung einer Routine kann als Aufruf eines Unterprogrammes (technologischer Teilarbeitsgang) betrachtet werden - wenn eben die Aufgabe schon gelöst ist. Dies läßt sich sprachunabhängig über die Anzahl der Parameter beschreiben.

- η_2^* Anzahl der Operanden, die als Parameter auftreten und
- η_1^* Anzahl der Operatoren, die als Aufruf veranlassen, was aber immer als η_2^* gesetzt werden kann.

In dieser Interpretation ist die minimale Größe der Routine (V^*) (des Teilarbeitsganges)

$$V^* = (2 + \eta_2^*) \cdot \text{lb} (2 + \eta_2^*).$$

Der Grad der "Ausformulierung" der Software bzw. des technologischen Teilarbeitsganges, deren Umfang durch das Verhältnis des realen zum potentiellen Quellencodumfang beschrieben werden kann, ist die algorithmische Tiefe

$$L = V^*/V.$$

Die von Halstead /8/ formulierte Hauptthese aus diesen Kenngrößen lautet für die Belange der Softwarebeurteilung, daß die Komplexität der Aufgabe, die Anzahl logischer Schritte bei der Formulierung in der Programmiersprache, proportional zu V und umgekehrt proportional zu L ist.

Für technologische Abläufe gilt analog, daß die Komplexität der algorithmierten Schritte im Ablaufmodell proportional zu V und umgekehrt proportional zu L ist. Bei der Wertung neuer Technologien, bei denen nach der Evolutionstheorie ein höheres Niveau der Komplexität und Organisiertheit gegenüber dem Status quo vorliegen soll, da dies ein Charakteristikum der Höherentwicklung ist, wäre somit z. B. eine Zunahme von V bzw. Abnahme von L nachzuweisen. Ferner kann aus den Kenngrößen L und V der mentale Aufwand E

$$E = V/L \quad \text{und}$$

der Programmieraufwand (Entwicklungsaufwand für die Technologie)
T

$$T = \frac{1}{S} \cdot E$$

abgeschätzt werden. Dabei ist $1/S$ ein Proportionalitätsfaktor.

wo?
Im Bild 9 sind die Halsteadwerte eines Softwarebeispiels (PASCAL-Funktion) berechnet.

Halstead beschäftigt sich auch mit der Frage der Umsetzung eines durchformulierten Vorganges in eine bestimmte Programmiersprache und bezeichnet dies als Komponentendesign. Dabei geschieht dies nicht nur aus der Sicht des Aufwandes, sondern aus der Sicht der zu erwartenden Fehler, die wiederum eine Funktion der Komplexität sind.

Es ist

$$\hat{B} = \text{Anzahl der bei einer Implementierung zu erwartenden Fehler.}$$

Für Softwareimplementierungen hat Halstead die Beziehung

$$\hat{B} = V/3\,000$$

gefunden.

Diese Fehler sind die üblicherweise als Programmierfehler bezeichneten.

Sieht man einen algorithmierten Ablauf (z. B. eine Technologie) in verschiedenen Ausführungsvarianten (unterschiedliche technologische Lösungen), dann bezeichnen die Halsteadwerte gewissermaßen die "Umständlichkeit", mit der ein technologischer Vorgang verschiedenartig ablaufen kann (algorithmiert ist).

Auch diese Kenngröße der Komplexität kann für die zielgerichtete zukunftsorientierte Technologieentwicklung eine hilfreiche Wertungsgröße sein.

4. Komplexität und Anlehnungen an die Rechenkomplexität

Die Theorie der Rechenkomplexität hat sich in jüngster Zeit zu einem Spezialgebiet entwickelt. Sie stellt die Ausgangsfrage

- Wieviele Rechenschritte benötigt ein Algorithmus?
- Wieviel (maximaler) Speicherplatz wird von ihm beansprucht?

Die Rechenschritte werden als Grundanweisungen nachfolgend verstanden. Auf die Frage des Speicherplatzbedarfes wird nicht weiter eingegangen.

Hat man einen unbekanntem Algorithmus vor sich oder befindet man sich auf der Fehlersuche, so ist es üblich, eine "Analyse durch Nachvollzug" zu praktizieren. Also einen Nachvollzug des Rechenganges, und dies hat schon etwas mit der Anzahl der Rechenschritte und der Komplexität zu tun. Hier sieht man auch den engen Zusammenhang zu graphentheoretischen Überlegungen. Denn ein solcher Rechendurchgang ergibt sich als Weg durch einen Ablaufgraphen, bei dem jede Kante mit der Anzahl von Grundoperationen bewertet ist, die nicht Steuerflußverzweigung oder Zusammenführungen sind. Die Weglänge ist letztendlich als Anzahl Knoten addiert mit der Gewichtssumme aus den durchlaufenden Kanten gleich der Anzahl Rechenschritte.

Auch diese Betrachtungsweise der Komplexität ist anwendbar für algorithmierte technologische Abläufe.

5. Komplexitätsbetrachtung unter Ansätzen der Logik

Es haben sich in der Logik Varianten einer logischen Komplexität herausgebildet, die sich untereinander bedingen und in den Hierarchien der Rekursionstheorie zusammengefaßt sind.

Algorithmierbarkeit von Technologie ?

Für Technologiebetrachtungen können bislang drei Ansätze der Komplexitätsbetrachtungen verfolgt werden, deren praktische Verwertung jedoch noch nicht ausgereift sind:

- Für die gesamte Logik ist bedeutsam die Frage der syntaktischen Struktur eines logischen Ausdruckes. So ist die klassische Logik selbst unterteilt in Aussagenlogik und Prädikatenlogik.
- Die Beweistheorie benutzt die Größe "Länge eines Beweises". Soll nun z. B. mit Hilfe einer Formel gezeigt werden, ob sie in einer Theorie beweisbar ist oder nicht, so ergeben sich in schnittfreien Kalkülen Beweisbäume, deren Tiefe ein Ausdruck der Komplexität ist.
- Neben bereits genannter Rechenkomplexität wird in der Logik weiter eine Hierarchiebetachtung durchgeführt bzw. ein konstruktiver Aufbau von Mengen aus logisch einfachen erzeugenden Elementen.

Es gibt bislang keine direkt greifbaren Methoden, daraus Maßzahlen der Komplexität zu entwickeln. Es liegt aber nahe, die Anzahl der für eine Verifikation notwendigen logischen Grundschritte als Komplexitätsgröße einzusetzen.

Mit der Zunahme des Einsatzes von algebraischen Spezifikationen, "logischen" Programmiersprachen und anderen logischen "höheren" Methoden (z. B. bei der Entwicklung wissensbasierender Systeme) eröffnet sich zweifelsohne auch für die Technologie die Möglichkeit, Überlegungen der skizzierten Art für Wertungen der Komplexität auszunutzen.

6. Zusammenfassung

Die objektive Notwendigkeit, die Softwareentwicklung und Technologieentwicklung eng zu verbinden, um effiziente Systeme zu schaffen, erfordert auch, gleiche Sprachregelungen und Wertungen zu benutzen.

Für die Kategorie Komplexität wurden unter Anwendung der Ähnlichkeitstheorie, Ansätze dargestellt, die zur Software- und Technologiebewertung in gleicher Weise benutzt werden können.

Derartige neue Betrachtungsweisen der Technologie werden für die evolutionäre Technologieentwicklung immer stärker an Bedeutung gewinnen.

Die Transporttechnologie ist aufgefordert, sich diesen Fragen progressiv zu stellen.

Literatur

- /1/ Hofmann, U.; Richter, K.-J.: Anwendung der Evolutionstheorie im Verkehrswesen, WZ der HfV (November 1987)
- /2/ Mc Call, James, A.: Factors in Software Quality; NTIS AD/A - 0049014; November 1977
- /3/ Mc Call, James A. et. al.: Software Quality metrics enhancements, Vol. 1; Rome Air Development Center, Griffiss Air Force Base; April 1980
New York
- /4/ Schick, George J.; Wolverson, Ray W.: Analysis of Computing Software Reliability Models; IEEE Transact. on Softw. Engineering; Vol SE - 4, No 2; March 1978
- /5/ Jourdon, Edward/ Constantine, Larry L.: Structured Design; Jourdon Press, New York; 1976
- /6/ Masing, W.: Handbuch der Qualitätssicherung; Hanser, München; 1980
- /7/ Boehm, B. w. et. al.: Characteristics of Software Quality; Elsevier, North Holland, Amsterdam; 1978
- /8/ Halstead, Maurice H.: A Theoretical Relationship Between Mental Words and Machine Language Programming; West Lafayette, Purdue University Computer Science Department; February 23, 1972