Datenstrukturen und höhere Programmiertechniken

von

Dr. Hartmut Noltemeier

Professor an der Universität Göttingen

Sammlung Göschen Band 5012

Walter de Gruyter

Berlin · New York · 1972

Die Reihe „Informatik" in der Sammlung Göschen umfaßt folgende Bände:

Einführung in Teilgebiete der Informatik. Von *L. Hieber.* 2 Bände
Digitale Rechenautomaten. Von *R. Klar.*
Analog- und Hybridrechner. Von *G. Gensch.* (In Vorb.)
Datenübertragung und -fernverarbeitung. Von *K. Oettl.* (In Vorb.)
Programmierung von Datenverarbeitungsanlagen.
Von *H.J. Schneider* u. *D. Jurksch.*
Datenstrukturen und höhere Programmiertechniken. Von *H. Noltemeier.*
Betriebssysteme I. Grundlagen. Von *E.J. Neuhold.* (In Vorb.)
Betriebssysteme II. Von *P. Caspers.* (In Vorb.)
Theorie und Praxis des Übersetzerentwurfs. Von *H.J. Hoffmann.* (In Vorb.)
Schaltwerk- und Automatentheorie. Von *C. Hackl.* 2 Bände
Graphentheorie für Informatiker. Von *W. Dörfler* u. *J. Mühlbacher.*
Einführung in die mathematische Systemtheorie. Von *F. Pichler.* (In Vorb.)
Formale Beschreibung von Programmiersprachen. Von *K. Alber.* (In Vorb.)
Angewandte Informatik. Von *P. Mertens.*
Information Retrieval. Von *O. Simmler.* (In Vorb.)
Programmiersprachen für die numerische Werkzeugmaschinensteuerung.
Von *U. Grupe.* (In Vorb.)

 – Satz: Fotosatz Prill, Berlin 15 – Druck: Mercedes-Druck, Berlin.
Printed in Germany

ISBN 3 11 003947 8

Vorwort

Unter Systemprogrammierung versteht man die zusammenfassende Darstellung derjenigen Techniken, die bei der Konzipierung und Erstellung von Systemprogrammen notwendig oder nützlich sind. Systemprogramme sind dabei die Teile eines datenverarbeitenden Systems, die dem Anwender die hardwaremäßigen Möglichkeiten des Systems möglichst vollkommen erschließen oder ihn von problemunspezifischen Arbeiten entlasten sollen: Steuerprogramme, Systemverwaltungsprogramme, Übersetzer, Dienstprogramme etc.

Der Inhalt des Bandes stellt eine gestraffte Einführung in die Grundlagen dieses Gebiets dar, die der Verfasser im Rahmen einer Vorlesung im Sommersemester 1971 an der Universität Karlsruhe für Studenten der Informatik und des Operations Research gegeben hat.

Besonderes Gewicht wurde dabei neben der Darstellung der im Rahmen der Systemprogrammierung üblichen Programmiertechniken auf die Beschreibung von Datenstrukturen und die Schilderung der wichtigsten Dateitypen gelegt, weil sie sowohl für den Ersteller von Systemverwaltungsprogrammen, von Übersetzern für Dialog- oder Datenbanksprachen u.ä. als auch für den anspruchsvolleren Anwender von großem Interesse sind.

Im Mittelpunkt der Ausführungen steht dabei der Begriff der Datei mit den Hauptkomponenten: logische Struktur, Zugriffsmöglichkeit, „Semantik". Zur Beschreibung der erstgenannten Komponenten erweisen sich graphentheoretische Begriffe als äußerst nützlich und wirkungsvoll. Der Verfasser hat daher die unmittelbar benötigten graphentheoretischen Hilfsmittel in § 1 vorangestellt, um den mit diesen Begriffen wenig Vertrauten Gelegenheit zu geben, ohne Rückgriff auf spezielle graphentheoretische Literatur die anschließenden Ausführungen zu verfolgen.

Da Systemprogramme vielfach unmittelbar Bezug nehmen müssen auf die Hardware des Systems, wird häufig eine hypothetische

Arbeitsmaschine eingeführt, um Programmiertechniken detailliert, aber ohne speziellen Bezug auf eine realisierte Maschine schildern zu können.

Der Verfasser konnte in diesem Zusammenhang darauf verzichten, weil er den Schwerpunkt auf die Darstellung allgemeiner Techniken legt, die jeder Systemprogrammierer vor der Konzipierung von Systemprogrammen kennen sollte – unabhängig von der speziellen Hardwareausstattung des Systems, mit dem er konfrontiert wird. Wo es erforderlich erscheint, sind Programmteile mit Algol-60-statements beschrieben.

Am Schluß jedes Paragraphen findet der Leser Übungsaufgaben, die nicht nur zur Kontrolle des Verständnisses dienen können, sondern zu intensiverem Studium einzelner Teilgebiete anregen sollen.

Besonderen Dank schulde ich den Herren Dipl.-Wi. Ing. Michael Bastian, Christian Fischer, Hermann Herz und Hans-Werner Six für ihre Mitarbeit sowie Frau Maier für die sorgfältige Schreibarbeit.

Göttingen, im Februar 1972

Hartmut Noltemeier

Inhalt

§ 1 Graphentheoretische Grundlagen

Zur Definition und Beschreibung spezieller Eigenschaften von Datenmengen („Dateien") erweisen sich graphentheoretische Begriffe als besonders nützlich und wirkungsvoll. Es sollen daher in diesem Abschnitt diejenigen Grundbegriffe der Graphentheorie vorangestellt werden, welche in obigem Zusammenhang unmittelbar relevant werden.

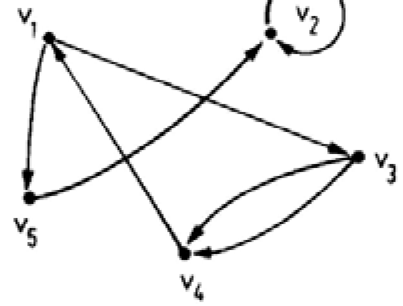
(1.1) *Def.* Ein Graph G ist ein Quadrupel $G = (V, R, \alpha, \omega)$ mit einer nichtleeren Menge V („Eckenmenge"), einer Menge R („Pfeilmenge") und zwei Abbildungen:

$\alpha : R \to V$ ($\alpha(r)$: „Anfangsecke" von $r \in R$)
$\omega : R \to V$ ($\omega(r)$: „Endecke" von $r \in R$).

Beispiel: $V = \{v_1, \ldots, v_5\}$, $R = \{r_1, \ldots, r_7\}$,

i	1	2	3	4	5	6	7
$\alpha(r_i)$	3	1	2	4	3	5	1
$\omega(r_i)$	4	5	2	1	4	2	3

Repräsentiert man die Ecken des Graphen ($v \in V$) durch Punkte in der Zeichenebene, die Pfeile ($r \in R$) durch gerichtete Verbindungslinien zwischen ihren Anfangs- und Endecken, so erhält man folgende geometrische Veranschaulichung:



(1.2) *Def.* Sei $G = (V, R, \alpha, \omega)$ ein Graph und $r_1, r_2 \in R$;

a) r_1 und r_2 sind *parallel* g.d.w.
$\alpha(r_1) = \alpha(r_2)$ und $\omega(r_1) = \omega(r_2)$.

b) r_1 ist eine *Schleife* g.d.w.
$\alpha(r_1) = \omega(r_1)$.

c) G ist ein *Graph ohne Parallelen* („G.o.P.") g.d.w. gilt: $\alpha(r_1) = \alpha(r_2), \omega(r_1) = \omega(r_2) \Rightarrow r_1 = r_2$.

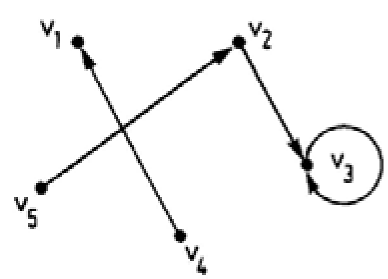
d) G ist ein *einfacher Graph* (kurz: G ist einfach) g.d.w. gilt: G ist ein G.o.P. und G besitzt keine Schleifen.

Graphen ohne Parallelen lassen sich einfacher formulieren, wenn wir die Tatsache benutzen, daß i.d.F. jeder Pfeil $r \in R$ durch seine Anfangs- und Endecke eindeutig bestimmt ist:

(*) $R \ni r \overset{1:1}{\longleftrightarrow} (\alpha(r), \omega(r)) \in V \times V$.

(1.3) *Def.* Ein Graph ohne Parallelen ist ein Paar $G = (V, R)$ mit einer nichtleeren Menge V („Eckenmenge") und einer Menge $R \subset V \times V$ (R: „Pfeilmenge").

Beispiel: $v = \{v_1, \ldots, v_5\}$, $R = \{(v_2, v_3), (v_4, v_1), (v_3, v_3), (v_5, v_2)\}$



$G = (V, R)$

Aus $G = (V, R)$ gewinnt man die ursprüngliche Definition dieses Graphen unmittelbar mittels (*).

(1.4) *Def.* (Globale Eigenschaften)
Sei G ein Graph o.P. und $u, v, w \in V$;

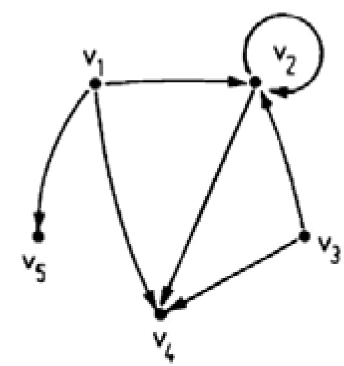
a) G ist *symmetrisch* g.d.w. gilt:
$(v, w) \in R \Rightarrow (w, v) \in R$.

b) G ist *antisymmetrisch* g.d.w. gilt:
$(v, w) \in R \Rightarrow (w, v) \notin R$.

c) G ist *transitiv* g.d.w. gilt:
$(u, v) \in R, (v, w) \in R \Rightarrow (u, w) \in R$.

d) G ist *vollständig* g.d.w. gilt:
$v \neq w, (v,w) \notin R. \Rightarrow (w,v) \in R$.

Beispiel:



G ist transitiv, aber nicht symmetrisch, nicht antisymmetrisch ($(v_2, v_2) \in R$) und nicht vollständig.

(1.5) *Def.* (Lokale Eigenschaften)
Sei $G = (V, R, \alpha, \omega)$ ein Graph und $v \in V$;

a) $N(v) := \{ w \in V \mid$ es existiert ein $r \in R$ mit $\alpha(r) = v, \omega(r) = w \}$
ist die *Nachfolgermenge* ($w \in N(v)$: „Nachfolger von v") von v.

b) $V(v) := \{ w \in V \mid$ es existiert ein $r \in R$ mit $\alpha(r) = w, \omega(r) = v \}$ ist die *Vorgängermenge* ($w \in V(v)$: *„Vorgänger von* v"*) von* v.

c) $g^+(v) := \| \{ r \in R \mid \alpha(r) = v \} \|$ ist der *Außengrad* von v,
$g^-(v) := \| \{ r \in R \mid \omega(r) = v \} \|$ ist der *Innengrad* von v,