



Lexikon Informatik und Datenverarbeitung

Herausgegeben von
Prof. Dr. Hans-Jochen Schneider,
UWE-Gruppe, Taucha

4., aktualisierte und erweiterte Auflage

R. Oldenbourg Verlag München Wien 1998

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Lexikon der Informatik und Datenverarbeitung / hrsg. von
Hans-Jochen Schneider. – 4., aktualisierte und erw. Aufl. -
München ; Wien : Oldenbourg, 1998
ISBN 3-486-24538-4
NE: Schneider, Hans-Jochen [Hrsg.]

Broschierte Ausgabe der 4. Auflage 1997

© 1998 R. Oldenbourg Verlag
Rosenheimer Straße 145, D-81671 München
Telefon: (089) 45051-0, Internet: <http://www.oldenbourg.de>

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Lektorat: Margarete Metzger
Herstellung: Rainer Hartl
Umschlagkonzeption: Kraxenberger Kommunikationshaus, München
Gedruckt auf säure- und chlorfreiem Papier
Gesamtherstellung: R. Oldenbourg Graphische Betriebe GmbH, München

Vorwort zur 4., völlig überarbeiteten und aktualisierten Auflage

Das Lexikon der Informatik und Datenverarbeitung verfolgt das ehrgeizige Ziel, den gesamten Bereich der Kerninformatik, der Angewandten Informatik und der Datenverarbeitung terminologisch abzudecken. Basis dieser Vorgehensweise ist, die Trennung zwischen theoretischer Hochschul-Informatik-Terminologie und praxisbezogener DV-Terminologie zu überwinden sowie der Angewandten Informatik einen breiten Rahmen einzuräumen. Interessierte Leser werden also sowohl Stichwörter über den Betrieb von Personal Computern (PC) und den Datenschutz, über Hardware, Software, Theorie der Informatik, Informationssysteme und Simulation finden als auch viele Einträge aus den Bereichen der Betriebs-, Produktions-, Umwelt-, Rechts- und Medizinischen Informatik. Die Terminologie der praktischen Datenverarbeitung deckt Begriffe vom Heimcomputer über Mikrocomputer und Minirechner bis zur Großdatenverarbeitung ab.

Dieses Ziel wäre für eine Person oder eine Gruppe von wenigen Personen nicht zu erreichen gewesen. Es ist dem Herausgeber gelungen, 199 fachlich kompetente Beitragende von dieser für das Fachgebiet lebensnotwendigen Aufgabe zu überzeugen. Ohne die fachliche Kompetenz, große Mühe und Geduld der Beitragenden wäre dieses Werk nicht entstanden. Der Herausgeber möchte sich bei den Beitragenden aus Wissenschaft, Industrie und öffentlicher Hand aus der Bundesrepublik und dem Ausland hiermit nochmals auf das herzlichste bedanken und hofft, daß das entstandene Werk und sein Weiterleben einen kleinen Ersatz für den großen geleisteten Einsatz darstellen. Die Liste aller Beitragenden findet sich nach den Hinweisen zur Benutzung des Lexikons.

Das Lexikon wendet sich entsprechend seiner Zielsetzung sowohl an die Praktiker in der Industrie, im Handel und der öffentlichen Hand als auch an Wissenschaftler, Lehrer an Gymnasien, Fachhochschulen und Universitäten, an Schüler, Studenten und Hobby-Computer-Fans.

Bei der inhaltlichen Gestaltung und Abstimmung unter den Beitragenden wurde ein völlig neuer Weg eingeschlagen. Die fachlichen Querbezüge, die Kontrolle der verwendeten Wortformen und die gesamte Verwaltung der Textbestände wurden dem computergestützten System zur semiautomatischen Lexikonerstellung TEAS (Terminologiedatenbank-Erfassungs- und Aufbereitungs-System) übergeben. Dieses System verwaltet nicht nur die Datenbank der gesamten Definitionstexte, sondern kennt auch alle Querbezüge zwischen den Definitionen und Stichwörtern und baut halbautomatisch eine Terminologie-Datenbank (Thesaurus) auf, in der alle in Definitionstexten vorkommenden Wortformen (inzwischen über 70 000) kontrolliert werden. Das System extrahiert und generiert ferner „auf Knopfdruck“ die notwendigen Lexikoneinträge, versieht sie mit zusätzlichen typographischen Drucksteuerzeichen und liefert die Disketten für die Lichtsatzmaschine. Diese neue Technologie erlaubt eine einfache Fortschreibung des Lexikons in der Zukunft. Meinen Dank möchte ich hiermit den Programmierern Herrn Hausherr sowie seiner Frau Ng aussprechen. Eine ausführlichere Beschreibung des Systems findet man am Ende des Lexikons. Das System TEAS wurde in den letzten Jahren weiterentwickelt und steht heute in Windows-Version zur Verfügung. Diese Version zeichnet sich durch eine Farb-Graphik-Schnittstelle aus und ermöglicht so dem ungeübten Benutzer ein Arbeiten mit dem System am Bildschirm, wie er es von einem Lexikon herkömmlicher Art gewöhnt ist.

Das Lexikon enthält auf über 1000 Seiten über 9000 deutsche Einträge im Hauptkapitel sowie über 3470 englische Einträge im englischen Index. Die Einträge verweisen auf eine Definition (Nebeneintrag) oder enthalten als Haupteintrag den Definitionstext. Zu jedem Stichwort findet man im allgemeinen das Teilgebiet, die englische Übersetzung, den Definitionstext, eventuelle Synonyme, Siehe-auch-Verweise und Abkürzungen. Eine genauere Beschreibung der Eintragstypen enthält das Kapitel „Hinweise für die Benutzung“.

Mein ganz besonderer Dank gilt Herrn Tilmann Hausherr, der in hervorragender Weise Konsistenzüberprüfungen vorgenommen hat und sich in akribischer Weise um all die kleinen Dinge kümmerte, ohne die ein solches Werk nicht leben kann. Eine große Hilfe bei den vielen Kleinigkeiten war Frau Wiltrud Weik, die mit viel Ausdauer und Kompetenz das Lexikon-Sekretariat führte.

Der Herausgeber und die Beitragenden fühlen sich beim bisher begangenen Weg durch die äußerst positive Resonanz auf die ersten drei Auflagen bestätigt. Die nun vorliegende *wesentlich erweiterte* vierte Auflage enthält Definitionen von über 30 neuen Beitragenden, die vorwiegend neue, bisher nicht behandelte Teilgebiete bearbeiteten, wie z. B. Desktop Publishing, Produktions-Informatik, Umwelt-Informatik, Wissensverarbeitung und Expertensysteme. Es liegt in der Natur eines solch komplexen und lebendigen Werkes, daß ständig Einzelheiten ergänzt oder geändert werden. Der Herausgeber ist all jenen zu Dank verbunden, die ihm ihre positiven und kritischen Beobachtungen beim Gebrauch des Lexikons mitteilen.

Der Herausgeber
Prof. Dr. Hans-Jochen Schneider
UWE-Gruppe
Taucha, Leipzig, Stuttgart

Inhaltsverzeichnis

Vorwort	V
Inhaltsverzeichnis	VII
Hinweise für die Benutzung	IX
Das Stichwort	IX
Aufbau eines Lexikoneintrages	X
Inhaltliche Querbezüge	X
Eine Grobklassifikation in Teilgebiete	X
Liste der häufig vorkommenden Stichwörter	XII
Beitragende	XV
Lexikon mit deutschen Eingängen und Definitionen	1
Lexikon mit englischen Eingängen und Verweisen auf deutsche Stichworte	1000
System zur semiautomatischen Lexikonerstellung	1057

Hinweise für die Benutzung

Das Stichwort

Die Auswahl der Stichwörter erfolgte ähnlich einem Delphi-Verfahren. Entsprechend der Bedeutung der Teilgebiete der Informatik und Datenverarbeitung wurden zuerst kompetente Fachleute zur Mitarbeit gewonnen, und diese wurden dann gebeten, 20 bis 40 wichtige Stichwörter mit Definitionen ihres Fachgebietes festzulegen. Nach diesem ersten Iterationsschritt wurden die ca. 70000 in den Definitionstexten vorkommenden Wortformen vom Herausgeber in Stichwörter und Banalwörter klassifiziert. Außerdem wurde die Terminologie in dem Sinne normalisiert, daß alle Flexionsformen (Plural, Genus, Deklination, Konjugation u. ä.) einen Verweis auf die Grundform erhielten und daß Synonyme, Wortfeldverweise (z. B. von Komposita auf Einzelwörter) und Nebeneinträge (z. B. von Synonymen auf Vorzugsbezeichnungen) festgelegt wurden. In der zweiten Iterationsphase wurden die neuen, durch Definitionstextanalyse gefundenen Stichwörter entweder definiert oder durch Neben- und/oder Wortfeldeinträge auf andere Definitionen zurückgeführt.

Synonyme Bezeichnungen verweisen auf denselben inhaltlichen Begriff. Bei den Vorzugsbenennungen steht der Definitionstext, die restlichen synonymen Bezeichnungen verweisen mit einem sogenannten **Nebeneintrag** auf diesen **Haupteintrag**. Z. B. erhält das Stichwort **Abtastregler** den Haupteintrag mit Definitionstext, und die synonymen Bezeichnungen **Regler**, **digitaler** und **digitaler Regler** verweisen mit → auf *Abtastregler*. Der Nebeneintrag wird weiterhin bei **Abkürzungen** und **Langformen** erzeugt (z. B. AM → *Amplitudenmodulation* und American Standard Code for Information Interchange → *ASCII*) und, wenn ein Stichwort auf die Definition eines anderen Stichwortes verweist, in der die Bedeutung des ersteren erklärt, plausibel gemacht oder durch den Kontext verdeutlicht wird (z. B. **Abtastzeitpunkt** → *Abtast- und Halteglied*). Der **Wortfeldeintrag** wurde eingeführt, um relevante Stichwörter mit untergeordneter Priorität nicht vollständig im Lexikon untergehen zu lassen. Für diese Wortfeldeinträge gibt es somit keine Haupt- und Nebeneinträge, aber an der Stelle ihres Auftretens in einem Definitionstext ist ein Verweis auf ein anderes Stichwort mit Eintrag vorhanden, welches mit dem Ausgangsstichwort in einem Bedeutungszusammenhang steht. Z. B. wurde das Stichwort „Impulsdauer“ dem Wortfeld von **Impuls** zugeordnet, da ein eigener Eintrag für Impulsdauer nicht als notwendig erachtet wurde, da schon eine Definition für Impuls vorhanden war. Im Text erscheint: **Impulsdauer** → *Impuls*. Gleichgeschriebene Wörter mit verschiedenen Bedeutungen – sogenannte **Homographen** – sind durch hochgestellte kleine Ziffern gekennzeichnet (Satz¹, Satz², Satz³).

Bei **mehrgliedrigen Stichwörtern** wird in der Regel das wesentliche Stichwort an den Anfang gesetzt und der Rest, durch Komma getrennt, angeführt. Die Ausgangsform erhält dann einen Nebeneintrag auf das normalisierte, mehrgliedrige Stichwort (z. B. **digitales Signal** → *Signal, digitales*).

Die Anordnung der Stichwörter geschieht streng nach dem ABC, wobei ä, ö, ü und ß wie die nicht umgelauteten Buchstaben eingeordnet wurden. Sonderzeichen wurden i. allg. so in die alphabetische Sortierung eingefügt, daß das Kernwort des Kompositums an den Anfang gestellt und das Sonderzeichen durch Komma abgetrennt und nachgestellt wurde (z. B. **3-Adreßbefehl** ist unter **Adreßbefehl, 3-** einsortiert).

Der Bindestrich erscheint, wenn die ersten Teile eine Abkürzung darstellen (z. B. **AD-Wandler**), wenn zwei Begriffe zusammengezogen wurden (z. B. **Ein-/Ausgabebefehl**), wenn der erste Teil ein Eigenname ist (z. B. **Aiken-Code**), wenn Teile in verschiedenen Sprachen auftreten (z. B. **Hash-Funktion**), wenn Gegenteiliges dargestellt wird (z. B. **Analog-Digital-Umsetzung**), wenn es sich um eine zusammengehörige Wortfolge handelt (z. B. **Ad-hoc-Anfrage**) und wenn es sich um ein sehr langes zusammengesetztes Wort handelt.

Aufbau eines Lexikoneintrages

Der Nebeneintrag besteht aus dem Stichwort und einem Pfeil auf das verwiesene Stichwort (z. B. **Abfallzeit** → *Schaltzeit*). Der Haupteintrag besteht aus dem Stichwort, dem Teilgebiet (siehe unten), der Übersetzung aus dem Deutschen ins Englische (manchmal auch umgekehrt, wenn der englische Begriff der eingeführte Sprachgebrauch ist), eventuell auftretende Synonyme, Abkürzungen, Langformen (Abkürzungen von), Siehe-auch-Verweise und Oberbegriffe. Anschließend folgt der eigentliche Definitionstext. Den Abschluß des Haupteintrages bildet der Name oder die Namen der für diese Definition zuständigen Beitragenden.

Inhaltliche Querbezüge

Im Definitionstext sind alle Stichwörter und alle Wortfeldverweise kursiv hervorgehoben. Der Leser weiß also immer sofort bei einem vorliegenden Definitionstext, auf welchen anderen Stichwörtern diese Definition aufbaut und wo er sich gezielt weiter informieren kann. Um die Texte nicht mit Kursivschriften zu sehr zu überlasten, wurde nur jeweils das erste Auftreten eines Stichwortes oder einer Wortform eines Wortfeldes durch Kursivschreiben hervorgehoben.

Außerdem sind sehr häufig vorkommende Wörter, wie z. B. **Änderung** oder **Datenverarbeitungsanlage**, gar nicht gekennzeichnet. Die Liste dieser Wörter findet der Leser auf den nächsten Seiten. Mit Hilfe dieser Verweisteknik kann der Leser im Lexikon navigieren und sich alle eventuell fehlenden Begriffsdefinitionen und Kontexte zusammensuchen, nur Banalwörter sind keinem Definitionstext zugeordnet.

Eine Grobklassifikation in Teilgebiete

Die nachfolgend aufgeführte Einteilung der Informatik und Datenverarbeitung in Teilgebiete ist als Grobklassifikation zur Unterstützung der Zusammenarbeit von Beitragenden und Herausgeber gedacht, sie erhebt keinerlei Anspruch auf Überlappungsfreiheit und Vollständigkeit. Sicher wird man sich in einigen Fällen streiten können, ob das Stichwort dem einen oder anderen oder eventuell sogar beiden Teilgebieten zugeordnet werden soll. Die Teilgebietsangabe soll dem Leser nur eine grobe Orientierung ermöglichen; die genaue Verfolgung der Bedeutungszusammenhänge gelingt ihm immer über die Kursivverweise in den Definitionstexten.

Allgemeines

- Allgemeines
- Grundlegende Definitionen

1. Theoretische Informatik

- Automatentheorie und Formale Sprachen
- Algorithmen und Komplexitätstheorie
- Logik
- Berechenbarkeit, Rekursive Funktionen
- Theorie und Semantik
- Graphentheorie
- Nebenläufige Prozesse

2. Software

- Datenstrukturen und -operationen
- Programmierung
- Programmierungsmethodik
- Übersetzerbau
- Betriebssysteme
- Programmierungstechnik (Software-Engineering)
- Rechnernetze und Verteilte Systeme

3. Hardware

- Rechnertechnologie
- Schaltwerke und Schaltnetze
- Mikroelektronik
- Rechnerorganisation
- Rechnerarchitektur
- Rechnerperipherie
- Datenübertragung
- Analog- und Hybridtechnik

4. Planung und Betrieb von Rechnersystemen

- Messung, Modellierung und Bewertung von Rechnersystemen
- Einsatzplanung und Installation
- Betrieb von Rechnersystemen

5. Angewandte Informatik – methodologische Aspekte

- Grundlagen von Informationssystemen (Datenbank-, Methodenbank- und Modellbanksysteme)
- Systemanalyse
- Modellierung
- Simulation
- Kognitive Methoden, Künstliche Intelligenz

- Wissensverarbeitung, Expertensysteme
- Interaktive Systeme, Mensch-Maschine-Dialog
- Computer Graphics und Computer Aided Design
- Linguistische Datenverarbeitung (Computerlinguistik)
- Information Retrieval
- Prozeßdatenverarbeitung
- Computergestützter Unterricht und Pädagogik
- Qualitätsmanagement
- Quantitative Methoden
- Telekommunikation
- Textverarbeitung
- Desktop Publishing
- Kryptographie

6. Angewandte Informatik – anwendungsbereichsbezogene Aspekte

- Informatik im Bankwesen
- Informatik in der Betriebswirtschaft (Betriebsinformatik)
- Informatik in der Fertigung (Produktionsinformatik)
- Informatik in Recht und öffentlicher Verwaltung (Rechtsinformatik)
- Informatik in der Medizin (Medizinische Informatik)
- Informatik im Umweltschutz (Umweltinformatik)
- Computer in der Kunst
- Computer in der Schule

7. Informatik und Gesellschaft

- Gesellschaftsrelevante Aspekte der Informatik
- Datenschutz
- Geschichte der Informatik

Liste der häufig vorkommenden Stichwörter

Um die Übersichtlichkeit und gute Lesbarkeit der Definitionstexte nicht zu beeinträchtigen, wurden die folgenden, häufig auftretenden Stichwörter, ihre Flexionsformen und die Wortformen aus ihren Wortfeldern im Definitionstext nicht kursiv gedruckt:

Ablauf	Area
ableitbar	Argument
Ableitung	arithmetisch
Adresse	Atom
Änderung einer Datenbank	Auftrag
Aktion	Ausdruck
Algorithmus	Bedienung
Anweisung	Bedingung
Anwendung	Befehl

Benutzer	Leitungsgebühren
Bild	Leitungssystem
Bilddarstellung	Maschine
Code	Maßzahl
Computer	Matchcode
computerunterstützt	mehrdeutig
Daten	Modell
Datenverarbeitung	Modellbegriff
Datenverarbeitungsanlage	Modulationsverfahren
digital	modulo
Diode	Monoid
display entity	Multitasking
Eigenschaft	Name
eindeutig	Netz
Eingabe	normalisieren
Eingang	Objekt
Entscheidung	Optimierung
Entschlüsselung	Organisation
Entwurf	Paar
Ereignis	Pfeil
erfüllbar	physikalisch
Fehler	Programm
Formel	Programmierung
Funktion	Programmsystem
Funktionseinheit	Prozedur
Genauigkeit	Prozeß
Gerät	Quellprogramm
Grad	Rechner
gültig	Regel
Halbordnung	Regelung
Identifikation	rekursiv
Index	resident
Indextabelle	Rolle
Indexzahl	Satz
Informatik	Schaltung
Information	Schaltung, gedruckte
Interpretation	Schnittstelle
Kabelrundfunk	Schrift
kanonisch	Seite
Kapazität	setzen
Kennzeichen	Sicherheit
Klasse	Signal
kognitiv	Spannung
Kommunikation	Speicher
Kommunikationsnetz	Speicherung
Kopf	Sprache
Kopie	Stelle
Koppler, akustischer	Steuerung
Ladung	Struktur

Hinweise für die Benutzung

Symbol

System

Term

Text

Tiefe

Transformation

Transistor

Übergang

Übersetzung

Übertragung

Variable

Vergleich

Verweis

Visibilität

Vollständigkeit

Weg

Wort

Zeichen

Ziffer

Zufallsgröße

Zufallsvariable

Zustand

Beitragende

Prof. Dr. R. Adamov
Zürich

Dr. K. Adena
Siemens Nixdorf Informations-
systeme AG, Paderborn

Dr. P. Albrich
SIEMENS AG, Erlangen

Dr. D. Altenkrüger
SIEMENS AG, München

Prof. Dr. W. Ameling
TH Aachen

Dipl.-Inform. J. Aurisch
Berlin

Dipl.-Betriebswirt G. Barkow
GBU, Hamburg

Prof. Dr. G. Barth
Daimler-Benz AG, Ulm

Prof. Dr. I. S. Bátori
Universität Koblenz-Landau

Dipl.-Math. P. Baumann
Universität Zürich

Prof. Dr. H. Beilner
Universität Dortmund

Dr. A. Bergmann
Detecon, Bonn

Prof. Dr. W. Bibel
TH Darmstadt

BIFOA Köln

Dr. H. Biller
SIEMENS AG, München

Prof. Dr. R. Bischoff
Fachhochschule Furtwangen

Prof. Dr. H. H. Bock
RWTH, Aachen

Prof. Dr. F. Bodendorf
Universität Nürnberg

Prof. Dr. K. H. Böhling
Universität Bonn

Dr. P. Bollmann
TU Berlin

Prof. Dr. W. Brauer
TU München

Hubert von Braun
München

Prof. M. Bues
Fachhochschule Furtwangen

Prof. Dr. H.-J. Bullinger
Universität Stuttgart

Prof. Dr. W. H. Burkhardt
Universität Stuttgart

Prof. Dr. V. Claus
Universität Oldenburg

Prof. Dr. H. Czap
Universität Trier

Prof. Dr. P. Dadam
Universität Ulm

Dr. I. Dahlberg
ISKO, Frankfurt

R. Deuse
ANT Nachrichtentechnik GmbH,
Radeberg

Dipl.-Ing. K. Eckert
Konstanz

Prof. Dr. W. Effelsberg
Universität Mannheim

Dr. Phil. H.-J. Ehlers
Stuttgart

Prof. Dr. J. Encarnação
Fraunhofer Arbeitsgruppe für
graphische Datenverarbeitung,
TH Darmstadt

Prof. Dr. A. Endres
München

Prof. Dr. E. Falkenberg
Universität Nijmegen,
Niederlande

Prof. Dr. K. Fellbaum
TU Berlin

Dr. H. Fetzer
Berlin

Prof. Dr. K. Fialkowski
UNIDO, Wien

Prof. Dr. H. Fiedler
GMD Bonn

Dr. P. Fleischhauer
Köln

Dr. T. Flik
TU Berlin

Beitragende

Dr. S. Florek
IBM Deutschland GmbH,
Sindelfingen

Prof. Dr. H.-P. Frei
Schweizer Bankgesellschaft, Zürich

Dr. D. Freiburg
Software AG, Darmstadt

Dr. Chr. Freksa
Universität Hamburg

Dr. U. Frenkel
Germering

Dipl.-Math. M. Fritsche
SIEMENS AG, München

Prof. Dr. G. Fuchs
Fachhochschule Fulda

Prof. Dr. N. Fuhr
Universität Dortmund

Prof. Dr. H. Gehring
Universität Hagen

Prof. Dr. S. Gerber
Leipzig

Prof. Dr. W. Giloi
GMD, TU Berlin

Prof. Dr. H. Gipper
Fachhochschule Aachen

Prof. Dr. W. Glatthaar
IBM Deutschland GmbH,
Stuttgart

Prof. Dr. H.-P. Godbersen
Technische Fachhochschule Berlin

A. Göbell
Kernforschungszentrum Karlsruhe

Prof. Dr. H. Göttler
Universität Mainz

Prof. Dr. P. Gorny
Universität Oldenburg

Prof. Dr. M. Gräf
Universität Tübingen

Prof. Dr. K.-D. Graf
Freie Universität Berlin

Prof. Dr. I. Grieger
Universität Stuttgart

Dr. N. Gronau
TU Berlin

Prof. Dr. R. Gunzenhäuser
Universität Stuttgart

Dr. S. Guse
Nürnberg

Prof. Dr. Ch. Habel
Universität Hamburg

Dipl.-Inform. R. Haberbeck
Berlin

Dr. E. Häußner
München

Prof. Dr. F. Haft
Universität Tübingen

Dr. H. Halfar
TU München

Dipl.-Ing., Dipl.-Kfm.
W. Hammann-Kloss
Neckargemünd

W.-R. Hansen
Darmstadt

Dr. R. Hartlep
Deutsche Post Consult, Berlin

Prof. Dr. D. Haupt
TH Aachen

Dipl.-Inform. T. Hausherr
Systemtechnik GmbH & Co. OHG
Berlin

Prof. Dr. H.-G. Hegering
Universität München

Prof. Dr. H.-D. Heike
TH Darmstadt

Prof. Dr. P. Hellwig
Universität Heidelberg

Prof. Dr. U. Herzog
Universität Erlangen-Nürnberg

Dipl.-Inform. H. Heß
Universität Saarbrücken

Prof. Dr. W. Hesse
Philipps-Universität Marburg

Prof. Dr. L. Hieber
Datenzentrale Baden-Württemberg,
Stuttgart

Dr. P. Hirschmann
Universität des Saarlandes

Prof. Dr. H.-J. Hoffmann
TH Darmstadt

Prof. Dr. R. Hoffmann
TH Darmstadt

Prof. Dr. W. Hummeltenberg
Universität Hamburg

Prof. Dr. M. Jarke
TH Aachen

Prof. Dr. W. Junginger
Universität der Bundeswehr,
Hamburg

Prof. Dr. A. Kaufmann
TH Gießen

H. Keutgen
GEI, Aachen

Dipl.-Inform. H.-B. Kittlaus
IBM Deutschland GmbH,
Böblingen

Prof. Dr. R. Klar
Universität Erlangen-Nürnberg

Prof. Dr. U. Kling
GMD, Darmstadt

Prof. Dr. W. Knödel
Universität Stuttgart

Prof. Dr. P. Koeppé
Freie Universität Berlin

Prof. Dr. E. Konrad
TU Berlin

Prof. Dr. H. Kopetz
TU Wien

Prof. Dr. H. Krallmann
TU Berlin

Prof. Dr. F.-L. Krause
Produktionstechnisches Zentrum
der Fraunhofer Gesellschaft,
TU Berlin

Prof. Dr. O. Kroschel
Universität Karlsruhe

Prof. Dr. R. Kuhlen
Universität Konstanz

Prof. Dr. E. Kuhlmann
TU Berlin

Dipl.-Ing. K. Kuhn
Micrologica, Bargteheide

Prof. Dr. K. Lagemann
Universität Hamburg

Prof. Dr. R. Lauber
Universität Stuttgart

Dr. L. Lehmann
Universität Hamburg

Prof. Dr. H. U. Lemke
TU Berlin

Prof. Dr. W. Lenders
Universität Bonn

Dr. H. Lenz
Universität Köln

Dr. M. Leszak
Philips Kommunikations
Industrie AG, Nürnberg

Prof. Dr. H. Liebig
TU Berlin

Prof. Dr. P. Lockemann
Universität Karlsruhe

Dipl.-Inform. D. Lohr
Waldenbuch

Dr. K. Graf v. Luck
Universität Hamburg

Dr. A. L. Luft
Universität Erlangen-Nürnberg

Prof. Dr. M. Lusti
Basel

Prof. Dr. Mätzel
Darmstadt

Prof. Dr. O. Manck
TU Berlin

Dr. L. Martiny
Paderborn

Dipl.-Ing. D. Matejka
ATM Computer GmbH, Konstanz

Prof. H. Meintzen
Allensbach

Dipl.-Inform. Chr. Mentzel
Makawao, Hawaii

Prof. Dr. P. Mertens
Universität Erlangen-Nürnberg

Prof. Dr. B. E. Meyer
Fachhochschule Heilbronn

Dr. K. H. Möller
Gröbenzell

Dr. H. Mönke
Stein an der Traun

Prof. Dr. K. Morik
Dortmund

Prof. Dr. H. Müller
Universität Erlangen-Nürnberg

Prof. Dr. P. Müller-Stoy
SIEMENS AG, München

Beitragende

Dr. R. Munz
SQL-Datenbank-Systeme GmbH,
Berlin

Prof. Dr. E. Neuhold
GMD, Darmstadt

Prof. Dr. B. Neumann
Universität Hamburg

Prof. Dr. H. Niemann
Universität Erlangen

Prof. Dr. J. Nievergelt
ETH Zürich

Dr. S.-A. Nilsson
Siemens Nixdorf-Informationssysteme AG, München

B. Nitzler
Aigaliers, Frankreich

Dr. R. Nowak
München

Prof. Dr. B. Page
Universität Hamburg

Prof. Dr. U. Pape
TU Berlin

Prof. Dr. H. A. Partsch
Universität Ulm

Prof. Dr. P. Pepper
TU Berlin

Prof. Dr. R. Popescu-Zeletin
GMD, TU Berlin

Prof. Dr. F. J. Radermacher
FAW, Ulm

Prof. Dr. B. Radig
Universität München

Prof. Dr. H. Reinermann
Hochschule für Verwaltungswissenschaften, Speyer

Prof. Dr. B. Reusch
Universität Dortmund

Prof. Dr. A. Reuter
Universität Stuttgart

Dr. G. Richter
GMD

Dr. J. Richter
Berlin

Prof. Dr. L. Richter
Universität Zürich-Irchel

Prof. Dr. M. M. Richter
Universität Kaiserslautern

Dr. G. Ringwald
Steinenbronn

Prof. Dr. C.-R. Rollinger
Universität Osnabrück

Prof. Dr. H. D. Rombach
Universität Kaiserslautern

Dipl.-Math. L. Rouette
Köln

Dr. G. Ruopp
ANT Nachrichtentechnik GmbH,
Backnang

Prof. Dr. H. Rzehak
Universität der Bundeswehr,
München

Dipl.-Kfm. Jörg Sander
Universität Saarbrücken

Prof. Dr. R. Schaßberger
TU Braunschweig

Prof. Dr. A.-W. Scheer
Universität Saarbrücken

Prof. Dr. J. Scherff
Fachhochschule Furtwangen

Prof. Dr. G. Scheschonk
C.I.T. GmbH, Berlin

Dr. G. Scheuernstuhl
Haessler-Software GmbH,
Schömberg

Prof. Dr. D. Schmid
Universität Karlsruhe

Prof. Dr. P. Schmitz
Universität Köln

Prof. Dr. H.-Jochen Schneider
UWE-Gruppe, Taucha

Prof. Dr. W. Schönfeld
IBM Deutschland GmbH, Heidelberg

Prof. J. Schürmann
Daimler-Benz AG, Ulm

Prof. Dr. M. Schulte-Zurhausen
Fachhochschule Aachen

Prof. Dr. H. Schweppe
Freie Universität Berlin

Prof. Dr. D. Seibt
Universität Köln

Dr. W. Seifert
AEG, Frankfurt

Dr. R. Soltysiak
Düsseldorf

Prof. Dr. G. Spur
TU Berlin

Prof. Dr. H. Stahl
TU Berlin

Prof. Dr. P. Stahlknecht
Universität Osnabrück

Prof. Dr. W. Steinmüller
Universität Bremen

Prof. Dr. F. Stetter
Universität Mannheim

Prof. Dr. G. Stiege
Universität Hildesheim

Prof. Dr. R. Stowasser
TU Berlin

Prof. Dr. Dj. Tavangarian
Universität Rostock

Prof. Dr. R. Thome
Universität Würzburg

Prof. Dr. R. Traunmüller
J.-Kepler-Universität, Linz

Dr. G. Tröller
Heinrich-Hertz-Institut, Berlin

Dr. H. Trost
Österr. Forschungsinst. für
Artificial Intelligence, Wien

Prof. Dr. C. Unger
FernUniversität Hagen

Dr. K. Voss
GMD, St. Augustin

Prof. Dr. K. Waldschmidt
Universität Frankfurt

Prof. Dr. H. H. Weber
TU Berlin

Dr. J. Wernicke
Berlin

Dr. H. K. Wild
Augsburg

Dipl.-Phys. E. Wildgrube
SIEMENS AG, München

Prof. Dr. H. Wojtkowiak
Gesamthochschule Siegen

Dr. Ch. Wolff
Universität Leipzig

Dr. H. Würges
SIEMENS AG, München

Prof. Dr. H. Zemanek
Wien

Prof. Dr. H. Zimmermann
Karlsbrunn

Prof. Dr. M. Zitterbart
Universität Braunschweig

Prof. Dr. K. Zuse †
Hünfeld

Abakus

Teilgebiet: Geschichte der Informatik

Rechenbrett, im Abendland zuerst erwähnt von Herodot (440 v.Chr.).

Ziffern werden durch Kugeln repräsentiert, die an einem Stab verschoben werden können. Die Kugeln hießen im Mittelalter Rechenpfennig oder Zahlpfennig. Ausdrücke wie „borge 1“ und „merke 2“ stammen von daher.

Der römische Abakus verwendet nicht neun Kugeln für die Ziffern von eins bis neun, sondern eine Gruppe von vier Kugeln für eins bis vier und eine einzelne Kugel, die fünf Einheiten zählt (bi-quinäre *Zahldarstellung*). Es sei daran erinnert, daß im alten Rom weder die Positionsschreibweise noch die Zahl 0 bekannt waren. Die heute käuflich zu erwerbenden Abaki sind Nachbildungen des chinesischen Abakus (600 v.Chr.), der zwei redundante Kugeln benützt, also fünf statt vier und zwei statt einer. *Knödel*

Abarbeitungsmechanismus → *Programmierung, deklarative*

Abbruch

Teilgebiet: Programmierungstechnik

Als Abbruch bezeichnet man das Unterbrechen eines laufenden Programms durch den Bediener (→ *Operateur*) der Datenverarbeitungsanlage, durch das *Betriebssystem* oder durch das Programm selbst. Ein Abbruch kann durch Datenfehler, durch Fehler im Programm oder in der Datenverarbeitungsanlage bedingt sein. Um bei umfangreichen Programmen das abgebrochene Programm nicht wieder beim Programmstart zu starten zu müssen, baut man an geeigneten Programmstellen *Wiederanlaufpunkte* ein. Ein abgebrochenes Programm kann dann beim letzten durchlaufenen Wiederanlaufpunkt neu gestartet werden. *Gehring*

Abbruchbedingung → *Abbruch*

Abbruchfehler → *Fehler bei der Anwendung von Theorien*

abduktiver Schluß → *Schluß, abduktiver*

abelscher zellulärer Automat → *Automat, zellulärer*

Abfallzeit → *Schaltzeit*

Abfrage bei Informationssystemen

Teilgebiet: Grundlagen von Informationssystemen

Siehe auch: Abfragesprache

In einem *Datenbanksystem* werden die Informationen durch Abfragen wiedergewonnen.

Die meisten Datenbanksysteme bieten zur Formulierung von Abfragen eigene Abfragesprachen an.

In *Auskunftssystemen* wird für den Benutzer oft eine Anzahl von parametrisierten (→ *Parameter*) Abfragen vordefiniert, so daß nur bestimmte Felder eines Bildschirms zur Formulierung der Abfrage ausgefüllt werden müssen. *Biller*

Abfragesprache

Teilgebiet: Grundlagen von Informationssystemen

query language

Siehe auch: Abfrage bei Informationssystemen

Eine Abfragesprache dient zur Wiedergewinnung von Information in *Datenbanksystemen*. Im engeren Sinne enthalten Abfragesprachen keine Elemente zur Datendefinition (→ *Datendefinitionssprache*) oder Datenänderung. Allerdings wird der Begriff Abfragesprache oft gleichbedeutend zu *Datenmanipulationssprache* verwendet. *Biller*

Abfragesystem → *Direktabfrage-System*

Abfragewunsch → *Direktabfrage-System*

Abgeordneteninformationssystem → *Parlamentsinformationssystem*

abgeschlossener Zweig → *Zweig, abgeschlossener*

abgesetzte Eingabe für Stapelbetrieb → *Stapelverfahrenverarbeitung*

abgesetzter Dateizugriff → *Dateizugriff, abgesetzter*

abgesetzter Zugriff → *Dateizugriff, abgesetzter*

Abhängigkeitsgrammatik, binäre

Teilgebiet: Linguistische Datenverarbeitung
binary dependency grammar

Binäre Abhängigkeitsgrammatiken sind *Grammatiken*, in denen höchstens zwei Elemente von einem Element abhängen. *Mönke*

ABK → *Optical Character Recognition-Leser*

Ablageverfahren → *Bürokommunikation*

Ablaufdiagramm → *Datenflußplan; Programmablaufplan*

Ablauf eines Programms

Ablauf eines Programms

Teilgebiet: Betrieb von Rechensystemen
program run

Eigentliche Ausführung eines Programms auf einem *DV-System*, wohingegen durch das Programm selbst nur die verschiedenen möglichen Formen des Ablaufs beschrieben werden. Im engeren Sinne gehört zum Ablauf eines Programms das Laden (\rightarrow *Lader*) des Programms in den *Hauptspeicher*, die Durchführung und Koordination der Eingabe und Ausgabe von Daten (über *Spooling*), die Fehlererkennung und Fehlerbehandlung durch ein parallel verfügbares *Ablaufsystem* und das ordnungsgemäße Beenden des Programms und die Freigabe der nicht mehr benötigten *Betriebsmittel*. Ergänzend dazu wird auch die Erfüllung der Aufgaben der *Ablaufsteuerung* häufig mit zum Ablauf des Programms gerechnet. Im einzelnen sind dies:

Berücksichtigung von *Prioritäten* beim Ablauf mehrerer Programme, die Koordination der Anforderungen der Programme und die Verwaltung und Verteilung von Betriebsmitteln. Graef

Ablaufgeschehen, Theorie des \rightarrow *Verkehrstheorie für Rechnersysteme und Netze*

Ablauflinie \rightarrow *Blockdiagramm zur Beschreibung von Programmabläufen*

Ablaufmodell

Teilgebiet: Grundlagen von Informationssystemen

Teil des Anwendungsmodells, der diejenigen Elemente des *Gegenstandsbereichs* enthält, die den Zusammenhang zwischen aktiven und passiven Elementen und zwischen verschiedenen aktiven Elementen herstellen.

Barkow; Hesse; Kittlaus; Scheschonk; von Braun

Ablauforganisation \rightarrow *Informationsorganisation; Organisation*

Ablaufplan \rightarrow *Blockdiagramm zur Beschreibung von Programmabläufen; Datenflußplan*

Ablaufplanung \rightarrow *Scheduling*

Ablaufsteuerung

Teilgebiet: Prozeßdatenverarbeitung
sequential control

Synonyme: Programmsteuerung; Taktsteuerung

Eine Steuerung mit zwangsläufig schrittweisem Ablauf, bei der das Weiterschalten von einem Schritt auf den programmgemäß folgenden abhängig von Weiterschaltbedingungen erfolgt.

Man unterscheidet zwischen zeitgeführter Ablaufsteuerung und prozeßabhängiger (\rightarrow *Prozeß*) Ablaufsteuerung. Bei der zeitgeführten Ablauf-

steuerung sind die Weiterschaltbedingungen nur von der Zeit abhängig. Bei der prozeßabhängigen Ablaufsteuerung sind die Weiterschaltbedingungen nur von Signalen der gesteuerten Anlage (*Prozeß*) abhängig. Kopetz

Ablaufsteuerung, prozeßabhängige \rightarrow *Ablaufsteuerung*

Ablaufsteuerung, zeitgeführte \rightarrow *Ablaufsteuerung*

Ablaufsystem

Teilgebiet: Programmierung
delivery system

Ein System (z.B. Arbeitsplatz-Rechner mit *Betriebssystem*) ausschließlich zum Ablauf, d.h. nicht zur Entwicklung von *Software*. Ablaufsysteme sind in der Regel kostengünstiger und leistungsstärker als Entwicklungssysteme (\rightarrow *Entwicklungssystem für Mikroprozessoren*), haben z.B. kleinere Platten (\rightarrow *Magnetplattenspeicher*) und Bildschirme, verfügen jedoch über keine komfortable *Software-Entwicklungsumgebung*. Altenkrüger

Ablauftheorie

Teilgebiet: Theorie der Semantik
trace theory

Insbesondere bei *nebenläufigen* (*parallelen*), nicht-sequentiellen Systemen bezeichnet man als Ablauf eines Prozesses eine abstrakte *Darstellung* eines Protokolls der Beobachtungen eines einzelnen hypothetischen Beobachters, der alle Aktionen (oder Ereignisse (\rightarrow *Ereignis in einem Datenverarbeitungssystem*)) des Prozesses, die er feststellen kann, nacheinander aufschreibt. Dabei werden simultane Aktionen (oder Ereignisse) in irgendeiner *Reihenfolge* notiert. Die Menge aller solcher Beobachtungen eines Prozesses in einem System stellt eine *operationale Semantik* dieses Prozesses dar, bei der zwischen Nebenläufigkeit und *Nichtdeterminismus* nicht unterschieden werden kann; die darauf beruhende Semantik-Theorie heißt Hoaresche Ablauftheorie.

Von A. Mazurkiewicz stammt eine Idee, wie man mit nur einer einzigen Beobachtungsfolge auskommen kann: Man gibt für das System an, welche der möglichen Aktionen (oder Ereignisse) unabhängig sind, d.h. in beliebiger Reihenfolge oder simultan auftreten können, dabei muß das System so beschaffen sein, daß sich diese *Unabhängigkeitsrelation* im Laufe der Zeit nicht ändert. Das ist z.B. bei *Petri-Netzen* der Fall. Bei dieser *Semantik* kann Nebenläufigkeit (als *Unabhängigkeit*) von *Nichtdeterminismus* unterschieden werden. Die auf dieser Idee beruhende Semantik-Theorie für nebenläufige Systeme heißt

Ableitungsbaum

Ableitungsbaum → *Ableitung in einer Grammatik*

Ableitungsnetz → *Ableitung in einer Grammatik*

Ableitungsregel → *Inferenzregel*

Ableitungssystem → *EKG-Auswertung, computergestützte*

Abmeldung eines Prozesses → *Rechenprozeß*

Abnahmeprüfung → *Funktionsprüfung*

Abnahmetest → *Funktionsprüfung*

Abnehmen

Teilgebiet: Programmierungstechnik

Abnehmen heißt, ein Arbeitsergebnis in die Verantwortung des Abnehmenden zu übergeben und den Ausführenden dadurch zu entlasten.

Anmerkungen:

1. Abnehmen ist damit eine in die Vergangenheit gerichtete Tätigkeit: Sie schließt einen Vorgang ab. Sie bedeutet keine Festlegung auf eine mögliche Verwendung des Arbeitsergebnisses.
2. Der Abnehmende ist häufig der *Auftraggeber*, der Ausführende der *Auftragnehmer* eines Herstellungsprozesses oder der Liefernde bei einem Software-Beschaffungsfall.
3. Im Abnahmevorgang getroffene Feststellungen über den Zustand des Arbeitsergebnisses, z.B. über angetroffene Mängel, werden im allgemeinen in einem Abnahmebericht festgehalten.
4. Häufig wird der Begriff des Abnehmens weitergefaßt. Vgl. z.B. VDI/VDE-Richtlinie 3690 (Abnahme von Prozeßrechnersystemen). *Fritsche*

Abrechnung der Maschinenzeit

Teilgebiet: Betrieb von Rechensystemen

Accounting

Synonyme: Benutzungsgebühren; Preise von DV-Ergebnissen

Zur Ermittlung der *Kosten* und ihrer Verursachung werden die von einem *DV-System* erbrachten Dienstleistungen gegenüber den Auftraggebern abgerechnet. Nur bei sehr einfachen Systemen ist diese Abrechnung unkritisch und für einen *PC* sogar unüblich, i.allg. ergeben sich Schwierigkeiten, den *Multiplexbetrieb* der CPU, die Aufteilung des *Hauptspeichers* unter mehreren Programmen bei gleichzeitiger dynamischer Speicherverwaltung (→ *Hauptspeicherverwaltung*), den *Parallelbetrieb* der Zentral- und Ein-/Ausgabeprozessoren, die Forderungen nach kurzen *Antwortzeiten* etc. mit in die Abrechnung in

„gerechter“ Weise einzubeziehen (*Abrechnungsroutine*).

Bei massiv *parallelen* Systemen wird vielfach nur die Belegungszeit der Rechenknoten, nicht aber deren tatsächliche Benutzung berücksichtigt.

Das Sammeln von Informationen über die Inanspruchnahme und zeitliche Belegung von *Betriebsmitteln* ist eine *Standardfunktion* der *Betriebssysteme* (*Logging*). Für die Auswertung dieser *Accounting-Daten* kommen unterschiedliche Programme je nach Zielsetzung zur Anwendung. Zum einen können die *Accounting-Daten* als Grundlage für eine *Kostenträgerrechnung* dienen, zum anderen können sie zur *Steuerung* des Ablaufs und zur *Belegungsplanung* herangezogen werden. *Graef; Haupt*

Abrechnungsroutine

Teilgebiet: Betriebssysteme
accounting routine

Installationsspezifische *Betriebssystem-Erweiterung*, die relevante Daten sammelt über den *Betriebsmittelverbrauch* (*CPU-Zeit*, *Speicherplatzbelegung*) einzelner Aufträge oder *Gespräche*. Aufgrund dieser Daten erfolgt ggf. eine *Weiterbelastung* der *DV-Kosten* an die einzelnen Benutzerabteilungen. *Endres*

Abrechnungssystem, elektronisches → *Banking-POS*

Abrufbetrieb

Teilgebiet: Rechnerarchitektur

Ein *Betrieb eines Rechensystems*, bei dem eine Zentraleinheit nach einer festgelegten Vorschrift von *Benutzerstationen* Daten abrufen. *Schneider*

Abrufphase

Teilgebiet: Rechnerorganisation
fetch cycle

Siehe auch: *Befehlsholphase*

Die Arbeitsphase des *Leitwerks*, in der das Leitwerk den Abruf eines Befehls oder *Operanden* aus dem Speicher steuert. *Klar*

Abschneiden → *Klippen*

Abschneideverfahren → *Truncation*

Abschnitt → *Auftrag*

abschnittsweise Verschlüsselung → *Verschlüsselung, abschnittsweise*

Absender einer Nachricht → *Quittung*

absolute Adresse → *Maschinenadresse*

absolute Häufigkeit → *Häufigkeit*

Absolutzeitimpuls

Teilgebiet: Prozeßdatenverarbeitung

Ein *Impuls*, der zu einem bestimmten Zeitpunkt (Uhrzeit) gesendet wird, in der Regel zum Anstoß eines *uhrzeitbezogenen* Auftrages. *Lauber*

Absolutzeituhr → *Prozeßperipherie*

Abstieg, rekursiver

Teilgebiet: Programmierung
recursive descent

Liegt die syntaktische (→ *Syntax von Programmiersprachen*) *Definition* einer *Programmiersprache* (etwa in der Form einer *Backus-Naur-Notation* (→ *Backus-Naur-Form*)) vor, so kann ein *Syntaxanalysealgorithmus* (→ *Sprachanalyse, syntaktische*) daraus nach folgendem *Verfahren* abgeleitet werden: Jedem nichtterminalen Symbol (→ *nichtterminales Zeichen*) (→ *Grammatik*) wird eine *Prozedur* zugeordnet, die das nächste *Teilstück* der *Eingabesymbolfolge* daraufhin untersucht, ob es aus diesem nichtterminalen Symbol ableitbar ist. Gibt es zu dem nichtterminalen Symbol mehrere *Produktionen*, so müssen die *Alternativen* im *Prozedurrumpf* (→ *Unterprogramm*) der *Reihe* nach abgeprüft werden. Innerhalb einer *Alternative* werden die terminalen Symbole mit den *Eingabesymbolen* verglichen und für die nichtterminalen die entsprechenden *Prozeduren* aufgerufen. Auf diese Weise wird der *Ableitungsbaum* von oben nach unten, also *top-down* rekonstruiert (→ *Analyse, syntaktische*). Das *Verfahren* heißt *rekursiv*, weil sich die konstruierten *Prozeduren* entsprechend der *rekursiven Definition* der *Programmiersprache* selbst aufrufen können. Es gibt *Programme*, die mit einer *kontextfreien Grammatik* als *Eingabe* die *Menge* aller für den *rekursiven Abstieg* nötigen *Prozeduren* generieren (→ *Compiler-Compiler*). *Günter*

Abstrakt

Teilgebiet: Information Retrieval
abstract; summary

Synonyme: *Kurzfassung; Referat*
Siehe auch: *Dokument*

Ein *Abstrakt* ist eine *Zusammenfassung* einer *vollständigen wissenschaftlichen Arbeit*. Es dient der *Schnellinformation* und soll die *Relevanzentscheidung* durch den *Benutzer* ermöglichen. *Abstrakte* werden vom *Autor* selbst oder im *Auswertungsprozeß* durch einen *Dokumentar* erstellt. Seit *Beginn* des 19. *Jahrhunderts* bis zur *Gegenwart* werden *fachspezifische Referateorgane* herausgegeben, in denen die nach *Hauptgruppen* geordneten *Abstrakte* neu erschienener *Veröffentlichungen* *zusammengefaßt* werden.

Heute sind solche *Referateorgane*, insofern sie *maschinenlesbar* vorliegen, die *Datenbasen* für weitere *Informationsverarbeitungsprozesse*. Man unterscheidet bei den *Abstrakt-Typen* zwischen *indikativen*, *deskriptiven* und *normativ-kritischen Abstrakten*.

In *technischen Fächern* werden häufig auch *Struktur-Referate* hergestellt. *Verfahren* des *automatischen Abstraktens* stützten sich in der *Vergangenheit* in erster *Linie* auf *Häufigkeitsinformationen* ab, mit deren *Hilfe* relevante *Sätze* *extrahiert* wurden. *Neuere Forschungsansätze* berücksichtigen stärker die *linguistische* (→ *Linguistik, mathematische*) *Texttheorie* und *Wissensrepräsentationstechniken* der *Künstlichen-Intelligenz-Forschung* (→ *Intelligenz, künstliche*).

Kuhlen; Zimmermann, H.

abstrakte Maschine → *Maschine, abstrakte*

abstrakter Datentyp → *Datentyp, abstrakter*

abstrakter Interpreter → *Interpretation bei Programmiersprachen*

abstrakte Simulation → *Simulation*

abstrakte Syntax → *Syntax, abstrakte*

Abstraktion

Teilgebiete: *Grundlegende Definitionen; Programmierungsmethodik abstraction*

Siehe auch: *Datentyp, abstrakter; Maschine, abstrakte*

Verfahren zur *Reduzierung* der *Komplexität* eines *Problems* durch *Trennung* der für bestimmte *Aspekte* der *Problemlösung* relevanten und irrelevanten *Details*. *Abstraktion* ist also stets *zielgerichtet*. Dieses *Ziel* bestimmt, welche *Details* eines gegebenen *Problems* hervorgehoben und für die *Problemlösung* als *relevant* eingestuft werden; die anderen (oft *unendlich vielen*) *Details* sind dann *irrelevant* - man sagt deshalb auch, daß beim *Abstrahieren* von *Details* abgesehen wird. So sind die *äußeren Maße* eines *Rechners* (→ *Datenverarbeitungsanlage, digitale*) und sein *Gewicht* für die *Ziele* des *Architekten* eines *Rechenzentrums* relevante *Aspekte*, für die *Ziele* der *Programmierer* dieses *Rechners* dagegen *irrelevant*. Und während sich ein *Programmierer* um die *programmiersprachen- und datenspezifischen Details* seiner *Programme* kümmern muß, möchten deren *Benutzer* davon i.a. mit *Hilfe* hierfür geeigneter *Spezifikationen* absehen und sich nur mit den *Details* eines *Programms* beschäftigen, die für *benutzungsspezifische Fragen* relevant sind: wenn sich *Programm-Benutzer* mit *Hilfe* hierfür geeigneter *Spezifikationen* nur für die

Wissensbestände interessieren, die Programme im Rahmen technischer Systeme maschinell verfügbar machen, so möchten sie sich um die anderen Details dieser Systeme nicht kümmern müssen. Von Abstraktion spricht man auch beim Aufbau wissenschaftlicher Terminologien (→ *Wissen*, objektives). Bei diesen sprachlich und methodisch anspruchsvollen Bemühungen wird zwischen Abstraktion und zielgerichteten, reduzierenden oder ausgrenzenden Problemlösungsansätzen schlechthin unterschieden: Im Rahmen der modernen Wissenschaftstheorie wird unter Abstraktion ein sprachlich-logisches Verfahren verstanden, das eine Rede über abstrakte Gegenstände (Abstrakta wie z.B. Begriffe, Daten, Symbole und Zahlen) methodisch nachvollziehbar einführt. Der Gebrauch von Ausdrücken wie z.B. „der Begriff Mensch“, „das Symbol ‚while‘“ oder „die Zahl 7“, die an der Subjektstelle abstrakter Aussagen wie z.B. „Die Zahl 7 ist ungerade“ vorkommen, aber nicht Eigennamen aufweisbarer Gegenstände sind, wird wissenschaftstheoretisch im Rahmen eines auf G. Frege zurückführenden Abstraktionsschemas explizit eingeführt: Wörter wie z.B. „Begriff“, „Symbol“ und „Zahl“ werden als Abstraktoren bezeichnet und zeigen eine Gleichbehandlung „konkreter“ Gegenstände bei abstrakten Aussagen an, wenn entsprechend diesem Abstraktionsschema definiert ist, über welche Gegenstände mit Hilfe des jeweiligen Abstraktors jeweils invariant (→ *Invariante*) geredet wird. Mit Hilfe dieses Abstraktionsschemas und dem dadurch definierten Gebrauch eines Abstraktors werden die „relevanten Details des Problems“ sprachlich explizit benannt und methodisch überprüfbar hervorgehoben sowie auf den hervorgehobenen „relevanten Details des Problems“ invariante Aussagen bezüglich der Gleichheit von Gegenständen ermöglicht, wodurch explizit zum Ausdruck gebracht wird, welche „Details eines Problems vorübergehend außer acht gelassen werden können“. Wenn wir beispielsweise sagen „die Zahl ‚4‘ ist gerade“, dann behaupten wir eine arithmetische *Aussage*, die insoweit abstrakt gemeint ist, als es auf die Repräsentation der Zahl durch eine bestimmte Ziffer nicht ankommt: Die Ziffern ‚4‘ und ‚IV‘ stellen in arithmetischen Aussagen die gleiche Zahl dar, d.h. der Abstraktor „Zahl“ zeigt eine invariante Redeweise an, der entsprechend dem Abstraktionsschema eine dazugehörige Gleichheitsbeziehung auf Ziffern zugrundeliegen muß. Diese ist durch ein Konstruktionsschema für Strichlisten sowie durch eine in arithmetischer Hinsicht definierte Gleichheit derjenigen Ziffern

gegeben, die konstruktionsgleiche Strichlisten bezeichnen. *Luft; Würges*

Abstraktionsebene → *Informationsmodellierung*

Abstraktionsstufe beim Programmwurf
Teilgebiet: Programmierungstechnik
refinement step

Programmebene beim funktionellen Programmwurf (→ *Programmwurf, strukturierter*), die unabhängig von darunter liegenden Abstraktionsstufen ausgetestet werden kann. *Schneider*

Absturz → *Abbruch*

Abtastalgorithmus → *Scan-line Algorithmus*

Abtasten, gerichtetes

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung
directed scan

Ein automatisches oder manuelles *Verfahren* für die rechnerinterne Generierung (→ *Synthese*) zweidimensionaler Daten. Das Abtasten eines *graphischen Datenträgers*, z.B. einer Zeichnungsvorlage, ist gerichtet, wenn eine Abtastordnung vorliegt. Diese *Ordnung* schreibt normalerweise ein zeilenweises, spaltenweises oder von der Form der graphischen Daten abhängiges Abtasten vor. *Encarnação; Lemke*

Abtaster → *Regelung*

Abtastfrequenz

Teilgebiet: Prozeßdatenverarbeitung
sampling frequency

Anzahl der Abtastungen bei einem Signal je Zeiteinheit. *Lauber*

Abtastordnung → *Abtasten, gerichtetes*

Abtastregelung → *Abtastregler*

Abtastregler

Teilgebiet: Prozeßdatenverarbeitung
sampled data feedback controller
Synonym: Regler, digitaler

Regler in einem *Regelkreis*, in dem nur zu diskreten Zeitpunkten die Reglereingangsgrößen erfaßt (abgetastet) und die *Stellgrößen* ausgegeben werden. *Lauber*

Abtast- und Halteglied

Teilgebiet: Analog- und Hybridtechnik
sample and hold device

Synonym: Sample-and-hold-Einheit

Ein Gerät, bei dem eine Eingangsgröße abgetastet wird und das eine Ausgangsgröße liefert, die gleich der Eingangsgröße im Abtastzeitpunkt ist.

Diese Größe wird bis zur nächsten Abtastung (→ *Abtasten, gerichtetes*) festgehalten.

Bei *elektronischen* Anwendungen wird ein Abtast- und Halteglied häufig einem Analog-Digital-Wandler (→ *Analog-Digital-Umsetzer*) vorgeschaltet, um dessen *Eingangsspannung* während der Umwandlungszeit konstant zu halten.

Albrich

Abtastzeit → *Analog-Digital-Umsetzung*

Abtastzeitpunkt → *Abtast- und Halteglied*

Abweichungsquadrate, kleinste → *Regression*

Abwickler → *Prozeßsteuerung*¹

AC → *Zugriffsbedingung*

Access Condition → *AC*

Account-File

Teilgebiet: Betrieb von Rechensystemen

Speicherdatei, die die von der *Abrechnungsroutine* gesammelten Informationen enthält. *Graef*

Accounting-Routine → *Abrechnung der Maschinenzzeit*

ACD

Teilgebiet: Telekommunikation

Siehe auch: ICM

System zur automatischen *Verteilung* von Anrufen auf Bearbeitertelefone. Eine ACD ist meistens eine integrierte Funktion einer Telefonanlage (TK-Anlage) oder eine mit der Telefonanlage gekoppelte Einheit. Mit der Verteilung der Anrufe wird eine *Statistik* erstellt, die die Verkehrsverteilung darstellt und für die Personaleinsatzplanung genutzt werden kann. *Kuhn*

8-4-2-1-Code → *BCD-Code*

ACIA → *Kommunikations-Schnittstellen-Baustein, asynchroner*

ACP → *Algebra of Communicating Processes*

ADA

Teilgebiet: Programmierung

Höhere *prozedurale Programmiersprache* zur Echtzeitdatenverarbeitung (→ *Echtzeitverarbeitung*). ADA wurde im Rahmen eines Wettbewerbs des amerikanischen Verteidigungsministeriums entwickelt. Es soll durch seine universelle Einsetzbarkeit sowohl im kommerziellen als auch im technisch-wissenschaftlichen Bereich dazu dienen, die *Wartungskosten* für *Software* möglichst gering zu halten. Ein ADA-Programm wird in der Regel auf einem (großen) Rechner entwickelt und *getestet* und kommt dann auf an-

deren (kleinen) Rechnern (→ *Datenverarbeitungsanlage, digitale*) zum Einsatz. ADA ermöglicht einen *modularen* Aufbau der Programme und getrennte Übersetzung der einzelnen *Module*, dies allerdings bei klarer *Definition* und *Überprüfung* der Schnittstellen. Zur Modularisierung stehen *Unterprogramme*, Pakete und Prozesse zur Verfügung. Pakete fassen zusammenhängende Daten und *Operationen* zusammen. Prozesse ermöglichen die *parallele Verarbeitung* (siehe auch *PEARL*). Der Name wurde zu Ehren von Augusta Ada Byron gewählt, einer Mitarbeiterin von Charles Babbage. *Burkhardt*

ADABAS

Teilgebiet: Grundlagen von Informationssystemen

ADABAS (Adaptierbares *Datenbanksystem*) ist ein relational-orientiertes Datenbanksystem, das von der *Software AG* entwickelt wurde und einfach mit der 4.-Generationssprache *NATURAL (Fourth Generation Language)* interagiert. Es stimmt mit relationalen Datenbanksystemen dahingehend überein, daß Informationen in *Tabelle*nform abgelegt werden, wobei Zeilen individuelle *Datensätze* repräsentieren und Spalten die zugehörigen Felder. Wie in relationalen Datenbanksystemen kann eine Verbindung zwischen verschiedenen ADABAS-Tabelleneinträgen über einen *Verbund* („join“) erfolgen. Im Gegensatz zu streng relationalen Datenbanksystemen erlaubt ADABAS jedoch die Verwendung von *multiplen Feldern*, *Feldgruppen* und *Periodengruppen*, womit die Einhaltung der *ersten Normalform* nicht mehr gefordert und ein Übergang zur *Non-First-Normal-Form (NF²)* möglich ist. *Heike*

Adaption → *System, selbstanpassendes*

adaptives Prozeßmodell → *Prozeßmodell, adaptives*

adaptive Wegeermittlung → *Wegeermittlung*

ADCON → *Adreßkonstante*

Addierer → *Addierwerk*

Addierer für Einskomentzahlen

Teilgebiet: Schaltwerke und Schaltnetze
adder for numbers in one's complement representation

Synonym: Addition von Einskomentzahlen

Durch eine n-stellige Einskomentzahl (→ *Einerkomplement*) $X = X_n X_{n-1} \dots X_1$ lassen sich die Werte von $-(2^{n-1} - 1)$ bis $2^{n-1} - 1$ darstellen. Ein Addierer für Einskomentzahlen bildet aus den Einskomentzahlen X und Y die

Addierer für Zweikomplementzahlen

Summe S in Einskomplementdarstellung. Wenn die Summe betragsmäßig kleiner als 2^{n-1} ist, dann berechnet sich die Summe nach der Beziehung

$$S = (X + Y + C_{n+1}) \bmod 2^n.$$

Darin bedeutet C_{n+1} den Übertrag in die Stelle $n+1$, wenn man ein n -stelliges Schaltnetz zur Addition von Dualzahlen verwendet. Der Übertrag C_{n+1} muß also auf die niederwertigste Stelle zurückgekoppelt werden und bewirkt in bestimmten Fällen eine Korrektur um $+1$ (Einskorrektur, end-around-carry).

Wenn die Summe betragsmäßig größer als $2^{n-1} - 1$ wird, dann erfolgt eine Bereichsüberschreitung, die wie bei der Addition von Zweikomplementzahlen (\rightarrow Addierer für Zweikomplementzahlen) ermittelt wird.

Wegen der notwendigen Einskorrektur ist die Realisierung der Addition von Einskomplementzahlen aufwendiger als von Zweikomplementzahlen (\rightarrow Zweierkomplement).

Hoffmann, R., Liebig

Addierer für Zweikomplementzahlen

Teilgebiet: Schaltwerke und Schaltnetze
adder for numbers in two's complement representation

Synonym: Addition von Zweikomplementzahlen

Durch eine n -stellige Zweikomplementzahl (\rightarrow Zweierkomplement) $X = X_n X_{n-1} \dots X_1$ lassen sich die Werte (abgekürzt WERT(X)) von -2^{n-1} bis $2^{n-1} - 1$ darstellen. Ein Addierer für Zweikomplementzahlen bildet aus den Zweikomplementzahlen X und Y die Summe S in Zweikomplementdarstellung, wenn die folgende Bedingung erfüllt ist:

$$-2^{n-1} \leq \text{WERT}(S) = \text{WERT}(X) + \text{WERT}(Y) \leq 2^{n-1} - 1.$$

Es läßt sich zeigen, daß sich die Summe S in Zweikomplementdarstellung auf eine Addition positiver Zahlen modulo 2^n zurückführen läßt:

$$S = (X + Y) \bmod 2^n.$$

Diese Funktion läßt sich durch ein n -stelliges Schaltnetz zur Addition von positiven Dualzahlen realisieren, indem der manchmal entstehende Übertrag C_{n+1} in der Stelle $n+1$ ignoriert wird. Wenn die Summe betragsmäßig zu groß wird, d.h. wenn

$$\text{WERT}(S) < -2^{n-1} \text{ oder} \\ \text{WERT}(S) > 2^{n-1} - 1$$

gilt, dann tritt der Fall der Bereichsüberschreitung (Überlauf, overflow) auf, weil n Stellen nicht mehr zur Darstellung ausreichen. Bei Verwendung eines n -stelligen Schaltnetzes zur Addition läßt sich die Überlaufbedingung OV nach folgenden Funktionen ermitteln:

$$OV = (X_n \sim Y_n) \wedge (C_{n+1} \neq S_n)$$

oder

$$OV = (\bar{X}_n \wedge \bar{Y}_n \wedge S_n) \vee (X_n \wedge Y_n \wedge \bar{S}_n).$$

Wenn eine Bereichsüberschreitung aufgetreten ist, dann ist die $(n+1)$ -stellige Summe durch $\bar{C}_{n+1} S_n S_{n-1} \dots S_1$ definiert. Die Addition in Rechenanlagen wird häufig in dieser Art durchgeführt, weil zur Realisierung ein n -stelliges Schaltnetz zur Addition genügt und keine Korrekturen, außer bei Bereichsüberschreitung, erforderlich sind.

Hoffmann, R., Liebig

Addierer mit Übertragsvorausberechnung

Teilgebiet: Schaltwerke und Schaltnetze
carry look ahead adder

Synonym: Carry-Look-Ahead-Addierer

Durch die Methode der Übertragsvorausberechnung gelingt es, die Additionszeit zu verringern. Beim Carry-Ripple-Addierer (\rightarrow Addierer mit Übertragsweiterleitung für Dualzahlen) werden die Überträge nacheinander durch die n rekursiven Gleichungen

$$(1) \quad C_{i+1} = (A_i \wedge B_i) \vee (A_i \vee B_i) \wedge C_i \\ i = 1, 2, \dots, n$$

berechnet. Realisiert man die Übertragsberechnung in dieser Form, dann werden zur Berechnung eines Übertrags pro Stufe zwei Gatterlaufzeiten (\rightarrow Schaltglied) T benötigt. Die gesamte Additionszeit beträgt damit $2nT + 1$. Die Anzahl der zu durchlaufenden Gatter läßt sich reduzieren, indem aufeinanderfolgende Gleichungen (1) ineinander eingesetzt werden. Durch Zusammenfassen der beiden Übertragungsgleichungen für $i=1$ und $i=2$ ergibt sich z.B.

$$(2) \quad C_3 = A_2 B_2 \vee (A_2 \vee B_2) A_1 B_1 \vee \\ C_3 = (A_2 \vee B_2) (A_1 \vee B_1) C_1$$

(Der Übersichtlichkeit halber wurde in dieser Beziehung das Zeichen für die UND-Verknüpfung zwischen den Variablen weggelassen.) Nach dieser Beziehung läßt sich C_3 ebenso schnell wie C_2 berechnen. Dafür hat sich der Realisierungsaufwand erhöht. Durch Zusammenfassen von drei Übertragungsgleichungen ergibt sich eine noch aufwendigere Beziehung, deren Realisierung es erlaubt, auch den Übertrag C_4 mit der gleichen Geschwindigkeit von $2T$ zu berechnen. In den so zusammengefaßten Übertragungsgleichungen wiederholen sich die Terme $G_i = A_i \wedge B_i$ und $P_i = A_i \vee B_i$ sehr oft, so daß sie zweckmäßigerweise als Hilfsfunktionen realisiert werden. G_i wird als Übertragserzeugung (carry generate) und P_i als Übertragsweiterleitung (carry propagate) bezeichnet. Der alte Übertrag C_i wird nach Gleichung (1) weitergeleitet, wenn $P_i = 1$ ist, und ein neuer Übertrag wird erzeugt, wenn $G_i = 1$ ist. Mit diesen Hilfsfunktionen lauten die Gleichungen für einen

Addierer mit Übertragsweiterleitung für Dualzahlen

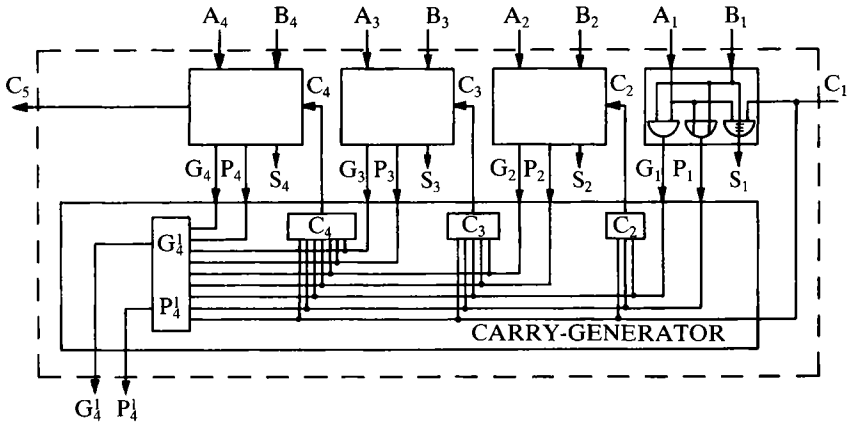


Bild a: 4-Bit-Addierer mit Übertragsvorausberechnung

4-Bit-Addierer (\rightarrow *Addierwerk*) mit Übertragsberechnung:

$$(3) C_2 = G_1 \vee P_1 C_1$$

$$(4) C_3 = G_2 \vee P_2 G_1 \vee P_2 P_1 C_1$$

$$(5) C_4 = G_3 \vee P_3 G_2 \vee P_3 P_2 G_1 \vee P_3 P_2 P_1 C_1$$

$$(6) C_5 = G_4 \vee P_4 C_4$$

$$(7) S_i = A_i \oplus B_i \oplus C_i$$

Der Übertrag C_5 kann nach der einfacheren Beziehung (6) gleichzeitig mit S_5 berechnet werden, die beide auf C_4 zurückgreifen.

Da der Aufwand zur Realisierung der Übertragungsfunktionen mit der dritten Potenz der Stellenzahl w steigt, werden im allgemeinen nur 4-Bit-Addierer mit vollständiger Übertragsvorausberechnung realisiert. Um z.B. einen 16-Bit-Addierer zu realisieren, können vier 4-Bit-Addierer mit Übertragsvorausberechnung nach dem Ripple-Carry-Prinzip hintereinandergeschaltet werden. In diesem Fall sollte die Berechnung von C_5 direkt von C_1 abhängig sein. Die Additionszeit beträgt dann das Vierfache des einzelnen 4-Bit-Addierers.

Diese Additionszeit läßt sich durch Einführen einer zweiten Carry-Generator-Ebene etwa auf die Hälfte reduzieren. Man bezeichnet die Realisierung der Funktionen (3), (4), (5) einschließlich der beiden Funktionen

$$(8) G'_4 = G_4 \vee P_4 G_3 \vee P_4 P_3 G_2 \vee P_4 P_3 P_2 G_1$$

$$(9) P'_4 = P_4 P_3 P_2 P_1$$

als Carry-Generator. Die Funktion G'_4 heißt als Übertragungserzeugung 1. Ordnung und P'_4 Übertragsweiterleitung 1. Ordnung. Schaltet man die Hilfsfunktionen 1. Ordnung auf einen weiteren Carry-Generator in einer zweiten Ebene, dann werden die Überträge C_5, C_9, C_{13} gleichzeitig mit erhöhter Geschwindigkeit berechnet. Gegenüber dem 16-Bit-Addierer nach Bild b) verringert sich die Additionszeit etwa auf die Hälfte.

Hoffmann, R., Liebig

Addierer mit Übertragsweiterleitung für Dualzahlen

Teilgebiet: Schaltwerke und Schaltnetze
carry ripple adder

Synonym: Carry-Ripple-Addierer

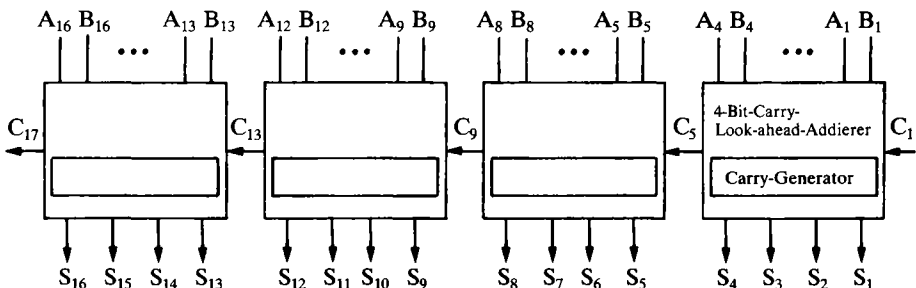


Bild b: 16-Bit-Addierer bestehend aus vier 4-Bit-Addierern mit Übertragsvorausberechnung

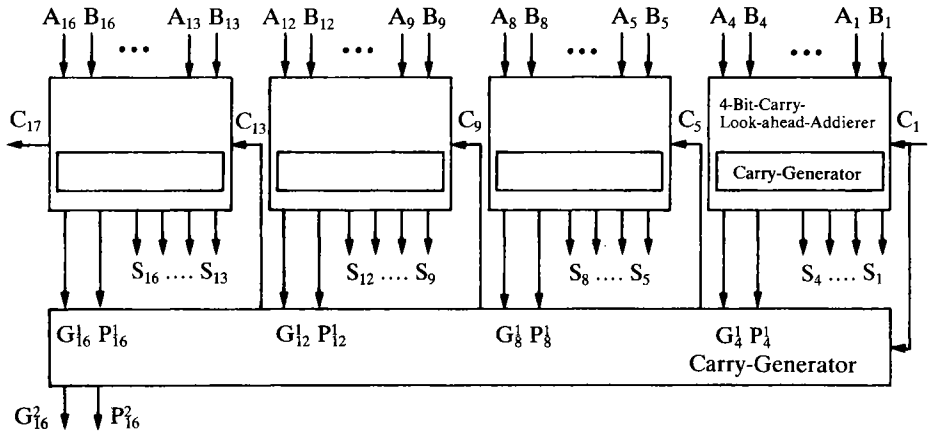


Bild c: 16-Bit-Addierer mit 2 Carry-Generator-Ebenen.

Ein *Schaltznetz* zur Addition von zwei n -stelligen *Dualzahlen* wird am einfachsten in Form von n hintereinander geschalteten *Volladdierern* realisiert. Durch jeden Volladdierer wird eine Stelle verarbeitet, wobei der *Übertrag* aus der vorhergehenden Stelle berücksichtigt wird. Der hineingehende Übertrag C_j (carry in) ist normalerweise gleich 0, außer bei der mehrfach hintereinander ausgeführten Addition oder anderen speziellen *Operationen*. Der herausgehende Übertrag C_n (carry out) ist gleich dem höchstwertigen Summenbit S_{n+1} , denn die Summe kann maximal den Wert $2(2^n - 1)$ annehmen. Er kann für eine Addition mit doppelter Wortlänge zwischengespeichert und dann als hineingehender Übertrag berücksichtigt werden. Im ungünstigsten Fall muß das Durchlaufen des Übertrags durch alle Stufen abgewartet werden, bis die Addition beendet ist. Dieser Addierer (\rightarrow *Addierwerk*) heißt deswegen auch carry-ripple adder. Die Additionszeit ist damit proportional zu der Anzahl der zu verarbeitenden Stellen. Sie läßt sich durch eine vorausschauende Berechnung der Überträge (Carry-Look-Ahead-Addierer (\rightarrow *Addierer mit Über-*

tragsvorausberechnung)) verringern.

Hoffmann, R., Liebig

Addierstufe \rightarrow *Halbaddierer*

Addierwerk

Teilgebiet: Schaltwerke und Schaltnetze

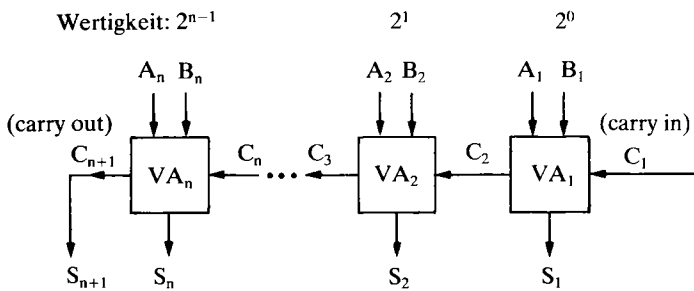
adder; addition unit

Synonym: Addierer

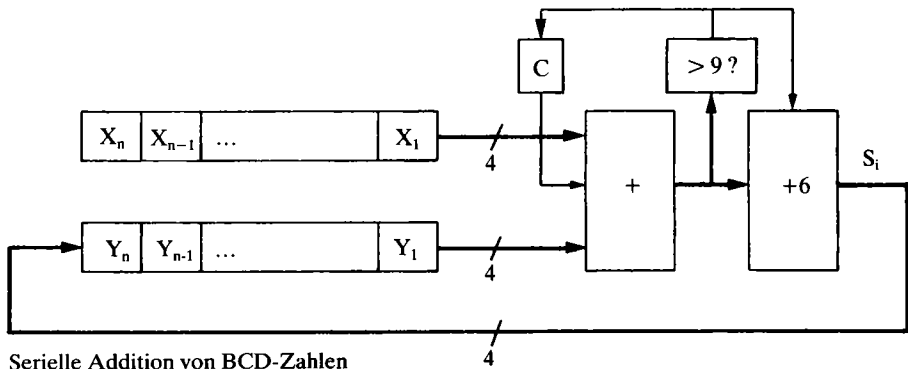
Addierwerke sind in der Lage, zwei oder mehr Summanden zu addieren und die Summe auszugeben. Sie können nach folgenden Merkmalen unterschieden werden:

- (1) Art der *Zahlendarstellung*
- (2) Anzahl der Summanden und deren *Definitionsbereich*
- (3) Form der *Realisierung* (z.B. *seriell, parallel* bzw. dem Algorithmus, auf dem die Realisierung beruht.

Die ersten beiden Unterscheidungsmerkmale lassen sich formal in der Abbildungsvorschrift zusammenfassen. Die gebräuchlichsten Arten der Zahlendarstellung sind *Dualzahlen*, *Zweikomplementzahlen* (\rightarrow *Zweikomplement*) und bi-



Additionsschaltznetz für Dualzahlen



Serielle Addition von BCD-Zahlen

när-codierte Dezimalzahlen (BCD-Zahlen (\rightarrow BCD-Code)), die entweder als Festkommazahlen (\rightarrow Festkommarechnung) oder als Gleitkommazahlen (\rightarrow Gleitkommadarstellung) (dargestellt durch Mantisse und Exponent) repräsentiert werden.

Hoffmann, R., Liebig

Addierwerk für Dezimalzahlen

Teilgebiet: Schaltwerke und Schaltnetze

addition unit for binary coded decimal numbers

Synonym: Addition von binär-codierten Dezimalzahlen

Die Addition von binär-codierten Dezimalzahlen (BCD-Zahlen (\rightarrow BCD-Code)) wird auf die Addition der einzelnen BCD-Ziffern zurückgeführt. Die BCD-Ziffern X_i und Y_i werden nacheinander dezimal (\rightarrow Dezimalsystem) unter Berücksichtigung der Überträge addiert. Die dezimale Addition wird im allgemeinen auf die *duale* Addition zurückgeführt, indem eine Korrektur durchgeführt wird, falls die Ziffernsumme >9 wird. Eine Ziffernsumme von 10 bis 15 stellt eine unerlaubte Codierung (Pseudotetrade (\rightarrow Pseudodezimal)) dar. Ziffernsummen >15 erzeugen einen Übertrag in die nächste Tetrade. Die *serielle* Addition läuft nach folgendem Algorithmus ab:

1. Übertragungsspeicher löschen (\rightarrow Operation) ($C:=0$).
2. Schritt 3 und 4 für alle m BCD-Ziffern ($i=1, \dots, m$) wiederholen.
3. Duale Zwischensumme bilden ($S_i := X_i + Y_i + C$).
4. Korrektur, falls Ziffernsumme >9 (if $S_i > 9$ then $S_i := S_i + 6$; $C := 1$ else $C := 0$).

Ähnlich wie die serielle Addition von Dualzahlen läßt sich die serielle Addition von BCD-Zahlen mit Hilfe von zwei Schieberegistern (4 Bit breit) und einem dezimalen Schaltnetz zur Addition realisieren.

Da die Realisierung der Abfrage ($>9?$) aufwendig ist, wird häufig eine Vereinfachung vorge-

nommen, indem $+6$ addiert und auf >15 (d.h. Übertrag in die nächste Tetrade) abgefragt wird. Entsteht kein Übertrag, so muß 6 wieder subtrahiert werden.

Der Algorithmus lautet dann für ein *paralleles Rechenwerk*:

1. In allen BCD-Ziffern X_i des ersten Summanden $+6$ addieren. Überträge zwischen den BCD-Ziffern können nicht auftreten.
2. Dazu den zweiten Summanden dual addieren ($S := (X_i + 66 \dots 6) + Y$). Alle Überträge zwischen den Tetraden (Gruppen zu 4 Bits, BCD-Ziffern) für den 3. Schritt merken.
3. Von jeder BCD-Stelle, für die kein Übertrag entstanden ist, 6 abziehen (durch Addition von $-6 = (1010)$ in Zweikomplementdarstellung (\rightarrow Zweierkomplement)). Dabei dürfen die entstehenden Überträge zwischen den Tetraden nicht weitergeleitet werden.

Ein Additionswerk (\rightarrow Addierwerk) für BCD-Zahlen eignet sich gleichzeitig zur Addition von positiven und negativen Zahlen in Zehnkomplementdarstellung. Der Wert der Summe darf dann zwischen $-50 \dots 0$ und $+49 \dots 9$ schwanken, ohne daß eine *Bereichsüberschreitung* auftritt. Eine Bereichsüberschreitung kann daran erkannt werden, daß bei gleichem Vorzeichen der Summanden durch die Addition ein Vorzeichenwechsel hervorgerufen wird. Das Vorzeichen ist implizit in der höchstwertigen BCD-Ziffer enthalten ($++$ für $0, 1, 2, 3, 4$ und $-$ für $5, 6, 7, 8, 9$).

Hoffmann, R., Liebig

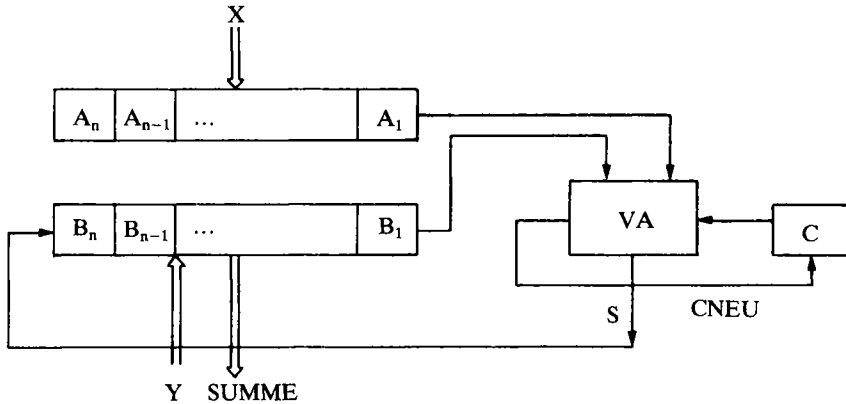
Addierwerk für Dualzahlen, serielles

Teilgebiet: Schaltwerke und Schaltnetze

addition unit for binary numbers; serial addition of binary numbers

Synonym: Addition von Dualzahlen, serielle

Ein *serielles Addierwerk* für Dualzahlen besteht aus zwei n -stelligen Schieberegistern, einem



Serielles Addierwerk für Dualzahlen

Volladdierer und einem Übertragungsspeicher. Es bildet nacheinander in n Schritten aus jeweils zwei Bits der Summanden und dem vorhergehenden Übertrag ein Summenbit und ein neues Übertragsbit. Der Ablauf der Summenbildung wird durch ein Steuerwerk nach folgendem Algorithmus gesteuert:

1. Wenn das Startsignal erfolgt, dann werden die Summanden in die Register übernommen und das Übertragsregister wird gelöscht ($A := X, B := Y, C := 0$).
2. Das Summenbit S und das Übertragsbit C_{NEU} werden mit Hilfe eines Volladdierers aus den niedrigstwertigen Registerbits A_1, B_1 und dem vorhergehenden Übertrag C gebildet.
3. Der neue Übertrag wird gemerkt ($C := C_{NEU}$), und die Registerinhalte werden um eine Stelle nach rechts geschoben. Gleichzeitig wird das Summenbit in das freiwerdende höchstwertige Bit des A-Registers geschrieben ($A := 0A_n \dots A_2, B := SB_n \dots B_2$).
4. Schritt 2 und 3 werden n -mal ausgeführt.
5. Die Summe steht im B-Register und der letzte Übertrag (carryout) im C-Register. Gehe nach 1. *Hoffmann, R., Liebig*

Addition von binär-codierten Dezimalzahlen
 → Addierwerk für Dezimalzahlen

Addition von Dualzahlen, serielle → Addierwerk für Dualzahlen, serielles

Addition von Einskomplementzahlen → Addierer für Einskomplementzahlen

Addition von Gleitkommazahlen
 Teilgebiet: Schaltwerke und Schaltnetze

addition of floating-point numbers
 Synonym: Addition von Gleitpunktzahlen

Durch Verwendung von Gleitkommazahlen (→ *Gleitkommadarstellung*) läßt sich ein großer Zahlenbereich überstreichen, ohne daß man die Stellung des Kommas berücksichtigen muß, wie es bei Festkommazahlen (→ *Festkommarechnung*) der Fall ist. Eine normalisierte Gleitkommazahl zur Basis b besteht aus der Mantisse mx und dem Exponenten ex und besitzt den Wert x gemäß der Beziehung

$$x = \begin{cases} mx \cdot b^{ex} & \frac{1}{b} \leq |mx| < 1 & \text{für } x \neq 0 \\ 0 \cdot b^{ex} & & \text{für } x = 0 \end{cases}$$

Um die Summe

$$s = ms \cdot b^{es} = mx \cdot b^{ex} + my \cdot b^{ey}$$

zu bilden, muß zuerst der kleinere Exponent an den größeren angeglichen werden. Dazu wird die Mantisse des Summanden mit dem kleineren Exponenten durch einen Shift (→ *Schiebeoperation*) um $|ex - ey|$ Stellen durch $b^{|ex - ey|}$ dividiert. Danach können die Mantissen addiert werden.

$$ms \cdot b^{es} = \begin{cases} (mx + my \cdot b^{-(ex - ey)})b^{ex} & ex \geq ey \\ (my + mx \cdot b^{-(ey - ex)})b^{ey} & ey > ex \end{cases}$$

Die nach der Addition entstehende Mantisse ist nicht notwendigerweise normalisiert. Um die Normalisierungsbedingung

$$1/b \leq |ms| < 1 \dots$$

zu erfüllen, muß die Mantisse um eine entsprechende Zahl von Schritten nach rechts oder links geschiftet und der Exponent erhöht oder erniedrigt werden. Die Normalisierungsbedingung läßt sich

bei einer *Darstellung* der Mantisse durch Betrag und Vorzeichen durch die Abfrage „höchstwertige Stelle ungleich Null“ realisieren.

Eine Zahlenbereichsüberschreitung tritt dann auf, wenn der Exponent durch die zur Verfügung stehenden Exponentenstellen nicht mehr dargestellt werden kann. Wird der Exponent zu groß, dann liegt ein Exponenten-Überlauf vor, wird er zu klein, dann liegt ein Exponenten-Unterlauf vor.

Hoffmann, R., Liebig

Addition von Gleitpunktzahlen → *Addition von Gleitkommazahlen*

Addition von Zweikomplementzahlen → *Addierer für Zweikomplementzahlen*

Additionsvektor → *Vektoradditionssystem*

Ad-hoc-Anfrage

Teilgebiet: Grundlagen von Informationssystemen

ad-hoc-query

Nicht vorher bestimmte Fragestellungen, wie sie z.B. in der Planung auftreten, führen zu spontanen, aktuellen Anfragen an die *Datenbank*. Dafür bieten *Datenbankverwaltungssysteme* spezielle *Abfragesprachen*, die sich am Terminalbenutzer (→ *Anwender*) ohne Programmierkenntnisse orientieren. Vom Benutzer werden aber gewisse Kenntnisse über Struktur und Inhalt der *Datenbank* verlangt.

Schneider

Adjunktion

Teilgebiet: Logik

disjunction

Synonym: Disjunktion

Extensionale aussagenlogische (→ *Aussagenlogik*) Verknüpfung (*Junktor*) mit dem Zeichen „ \vee “, die dem umgangssprachlichen „oder“ im nicht ausschließenden Sinn entspricht. Die *Aussage* $A \vee B$ (lies „A oder B“) ist wahr, wenn mindestens eine der Aussagen *A*, *B* wahr ist. Dies wird durch die folgende *Wahrheitstafel* formal beschrieben:

A	B	$A \vee B$
w	w	w
w	f	w
f	w	w
f	f	f

Statt „ \vee “ wird in älterer Literatur auch „+“ verwendet.

Müller

Adreßbestimmung

Teilgebiet: Programmierung

address calculation

Synonyme: Adressenrechnung; Adreßrechnung; Adreßumsetzung

Adreßbestimmung bezweckt, aus einer im Programm verwendeten Kennzeichnung eines Speicherplatzes (*Speicherzelle*) oder eines zusammenhängenden *Speicherbereichs* die dazugehörige *Maschinenadresse* zu ermitteln, mit der ein *Operand* angesprochen werden bzw. ein *Sprungbefehl* ausgeführt werden kann. Die Adreßbestimmung erfolgt in Stufen.

Wir unterscheiden (1) vorgezogene Adreßbestimmung und verstehen darunter alle Maßnahmen, die von einem *Interpreter*, *Assemblierer* oder *Übersetzer* vorgenommen werden können bzw. müssen, um *symbolische Adressen* in *numerische Adressen* umzuwandeln, und (2) eigentliche, zurückgestellte Adreßbestimmung (siehe unten). *Numerische Adressen* sind im ersteren Fall (im allgemeinen) aus mehreren adressierungsrelevanten Angaben zusammengesetzt, die den *Adreßteil* von Befehlen (*Befehlsformat*) bilden können, insbesondere *Basis-/Segmentangabe*, *Indexangabe*, *Angabe eines Adressenversatzes*. Zum Vorgehen in der ersten Stufe siehe auch *Bezeichner*, *Vereinbarung*, *Variablenvereinbarung*.

Eine zweite Stufe der Adreßbestimmung geschieht durch den *Binder* und/oder *Lader*. Es werden u.U. Änderungen an adressierungsrelevanten Angaben vorgenommen, wenn z.B. mehrere Module (→ *Modul²*) in eines zusammengefaßt werden - typischerweise Änderungen am *Adressenversatz* - oder die Ladeadresse von Modulen bestimmt wird (*Adreßkonstante*).

Nun steht jedenfalls eine maschinenbezogene, nicht mehr in irgendeiner Weise symbolische, sondern rein numerische Adresse für die eigentliche Adreßbestimmung zur Verfügung; diese geschieht (in der Regel) durch *Bearbeitungsschritte* in der *Rechner-Hardware*.

Die Adreßbestimmung ist abgeschlossen, wenn die Adresse elementar und als *Maschinenadresse* brauchbar ist.

Andernfalls schließen sich weitere Stufen an:

- Es kann sich um eine *indizierte Adresse* handeln; ein Indexwert (→ *Index* (*sog.*) in der *Textverarbeitung*), meist aus einem *Register* („*Indexregister*“) entnommen, geht in die Adreßbestimmung ein.
- Es kann sich um eine *Basis-/Relativadresse* handeln; der in vorangehenden Stufen ermittelte *Adressenversatz* wird als *Relativadresse* aufgefaßt und zur *Basisadresse* („*Basisregister*“) hinzuaddiert.
- Eine (heute gebräuchliche) Variante davon ist eine *segmentierte Adressierung*.

Adreßbestimmung, register-indirekte

Wenn der vorgesehene Rechnertyp (\rightarrow *Datenverarbeitungsanlage, digitale*) das Verfahren der *virtuellen* Adressierung verwendet, handelt es sich bei der insoweit bestimmten Maschinenadresse um eine Adresse im *virtuellen Adreßraum*. Zur Umwandlung in die Adresse im realen Adreßraum schließt sich eine weitere Adreßbestimmungsstufe an.

Bei der bestimmten Maschinenadresse kann es sich um eine *indirekte Adresse* handeln. Ein weiterer Sonderfall ist die Adressierung in einem Assoziationspeicher (\rightarrow *Assoziativspeicher*).

Hoffmann, H.-J.

Adreßbestimmung, register-indirekte

Teilgebiet: Programmierung
register indirect addressing

Synonym: Register-indirekte Adressierung

Eine Variante der *Adreßbestimmung* für eine *indirekte Adresse*, bei der die im Adreßteil des Befehls gespeicherte Adresse ein *Register* zur Entnahme der zweiten Adresse auswählt (Gegensatz: *Speicher-indirekte Adreßbestimmung*).

Hoffmann, H.-J.

Adreßbestimmung, speicher-indirekte

Teilgebiet: Programmierung
memory indirect addressing

Synonym: Speicher-indirekte Adressierung

Eine Variante der *Adreßbestimmung* für eine *indirekte Adresse*, bei der die im Adreßteil des Befehls gespeicherte Adresse eine *Speicherzelle* im *Hauptspeicher* zur Entnahme der zweiten Adresse auswählt (Gegensatz: *Register-indirekte Adreßbestimmung*).

Hoffmann, H.-J.

Adreßbus

Teilgebiete: Rechnerarchitektur; Schaltwerke und Schaltnetze
address bus

Als Adreßbus wird der Teil eines Verbindungs- oder Sammelschienensystems bezeichnet, der die Adreßsignale *parallel* (\rightarrow *parallel²*) zwischen den Funktionseinheiten überträgt, die mit dem *Bus* gekoppelt sind. Er verfügt bei Mikrorechnern (\rightarrow *Mikrocomputer*) in der Regel über 32 oder 64 parallele Signalwege, die *unidirektional* vom *Prozessor* als aktive Funktionseinheit zu passiven Funktionseinheiten (Speicher, Peripherie (\rightarrow *Peripheriegerät*)) übertragen. Sind mehrere aktive Funktionseinheiten vorhanden (z.B. Mehrprozessorsystem (\rightarrow *Multiprozessorsystem*), *DMA-Kanal*), so müssen diese mit *Treibern* in *Dreizustandslogik* bzw. *Treibern mit offenem Kollektor* realisiert sein (siehe auch *Datenbus*).

Nilsson

Adresse, absolute \rightarrow *Maschinenadresse*

Adresse, buchstäbliche \rightarrow *Operand an Adreßposition*

Adresse, direkte

Teilgebiet: Programmierung
direct address

Adresse, mit deren Hilfe ein *Operand* bei einer *Befehlsausführung* (direkt) angesprochen wird. Gegensatz: *indirekte Adresse*. Direkte Adressierung ist der normale Fall. Ein *direkt adressierter Operand* wird über eine direkte Adresse angesprochen.

Hoffmann, H.-J.

Adresse, echte \rightarrow *Maschinenadresse*

Adresse, effektive \rightarrow *Maschinenadresse*

Adresse, elementare

Teilgebiet: Programmierung

Eine Adresse, die keine *zusammengesetzte Adresse* ist. Falls sie eine *numerische Adresse* ist (und keine *Adreßwiederholung* erfolgt), muß sie im vorgesehenen *Adreßraum* liegen.

Hoffmann, H.-J.

Adresse, indizierte \rightarrow *Adresse, indizierte*

Adresse, indirekte

Teilgebiet: Programmierung
indirect address

Synonym: Adresse einer Adresse

Adresse, mit deren Hilfe zunächst ein Speicherplatz angesprochen wird, der seinerseits eine zweite Adresse enthält, mit deren Hilfe dann ein *Operand* für eine *Befehlsausführung* angesprochen wird. Gegensatz: *direkte Adresse*. Man kennt auch eine *indirekte Adressierung* in mehreren Stufen. Indirekte Adressierung ist entweder technisch realisiert (durch eine *binäre Aussage* im Adreßteil des Befehls; direkt/indirekt) oder durch zwei Befehle, wobei der erste die *Operandenadresse* in ein für Adressierungszwecke (\rightarrow *Adressieren*) benutzbares *Register* (*Indexregister*) lädt. Ein *indirekt adressierter Operand* wird durch eine indirekte Adresse angesprochen.

Hoffmann, H.-J.

Adresse, indizierte

Teilgebiet: Programmierung
indexed address

Zusammengesetzte Adresse, mit deren Hilfe Speicherplätze in vorgegebener Anordnung (meist nebeneinanderliegend, um gleichbleibenden *Adressenversatz* getrennt), z.B. zur Aufnahme eines Vektors der Reihe nach angesprochen werden können. In der zusammengesetzten Adresse findet sich eine Angabe für die *Basis-*

adresse der Anordnung und eine Angabe (meist eines *Registers*) für den laufenden Index. Der Index wird beim Durchlaufen der Reihe entsprechend dem jeweiligen Adressenversatz verändert (*Indexregister*). Es kann also mit einem Befehl, dessen *Adreßteil* der Indexmodifikation unterworfen ist, in einer (Programm)schleife der Reihe nach jeder zur Anordnung gehörende Speicherplatz angesprochen werden. Hoffmann, H.-J.

Adresse, literale → *Operand an Adreßposition*

Adresse, logische

Teilgebiet: Programmierung
logical address

(Nicht sehr gebräuchliche) Bezeichnung der virtuellen Adresse, wenn der *virtuelle Speicher* nicht auf den *Hintergrundspeicher* ausgedehnt ist, sein *Adreßraum* also kleiner oder höchstens gleich dem des tatsächlich verwendeten Arbeitsspeichers (→ *Hauptspeicher*) ist (bei *Swapping*). Hoffmann, H.-J.

Adresse, numerische

Teilgebiet: Programmierung
numerical address

Eine Adresse, deren Identität durch eine Zahlenangabe charakterisiert ist. Hoffmann, H.-J.

Adresse, physikalische → *Maschinenadresse*

Adresse, relative → *Relativadresse*

Adresse, strukturierte → *Adresse, zusammengesetzte*

Adresse, symbolische

Teilgebiet: Programmierung
symbolic address
Synonym: Adreßsymbol

Eine Adresse, deren Identität durch eine frei eingeführte Angabe charakterisiert ist. Sie muß (1) (bei *höheren Programmiersprachen*) aus einer *Vereinbarung* abgeleitet bzw. (2) (bei maschinenorientierten (→ *Maschinenabhängigkeit*) *Programmiersprachen*) durch ein definierendes Auftreten der Angabe als *Marke* (= Symboldefinition) einem Speicherplatz zugeordnet und (3) der *Adreßbestimmung* unterworfen werden, bevor sie als *Maschinenadresse* dienen kann. Die Angabe kann aus einer Zahl bestehen (*numerische Adresse*), meist wird allerdings eine *alphanumerische*, den Inhalt des durch die Adresse angesprochenen Speicherplatzes erklärende Angabe verwendet (z.B. ZAEHLER1). Wird unter der Adresse ein konstanter Wert angesprochen, kann für diesen eine Standardbezeichnung als Angabe dienen. In höheren Programmiersprachen übernehmen die dort verwendeten Namen (→ *Be-*

zeichner) für konstante wie für *variable* Werte auch die Rolle der symbolischen Adresse.

Hoffmann, H.-J.

Adresse, tatsächliche → *Maschinenadresse*

Adresse, wahre → *Maschinenadresse*

Adresse, wertbestimmende → *Adresse, symbolische*

Adresse, zusammengesetzte

Teilgebiet: Programmierung
compound address
Synonym: Adresse, strukturierte

Erfolgt die *Adreßbestimmung* nicht durch eine einstellige Abbildung oder sind bei einer mehrstelligen Abbildung die Bezugsadressen nicht dem Kontext oder einer Voreinstellung (→ *Default-Wert*) bzw. durch *Adreßwiederholung* zu entnehmen, muß der *Adreßteil* eines Befehls mehrere Angaben enthalten. Diese bilden dann eine zusammengesetzte Adresse. Beispiel: Adreßteil enthält *Registerangabe*, aus dem *Basisadresse* entnommen wird, *Registerangabe*, aus dem Adressierungsindex (→ *Adresse, indizierte*) entnommen wird, und *Adressenversatz*. Gegensatz: *elementare Adresse*. Hoffmann, H.-J.

Adresse einer Adresse → *Adresse, indirekte*

Adresse eines Speicherplatzes

Teilgebiete: Programmierung; Rechnerorganisation
address

Synonym: Speicheradresse
Siehe auch: Adresse, direkte

Ein bestimmtes Wort oder Teil eines Worts, meist eine Zahl, aber auch eine andere Angabe zur Kennzeichnung eines Speicherplatzes oder eines zusammenhängenden *Speicherbereichs*, der durch ein Programm als Einheit ansprechbar sein soll; technisch auch zur Kennzeichnung einer *Speicherzelle* bzw. einer Funktionseinheit. Wird keine Zahl vermerkt, die mit der durch den technischen Aufbau des Speichers fest vergebenen Kennzeichnung der anzusprechenden Speicherzelle, d.h. ihrer *Maschinenadresse* übereinstimmt, muß eine *Adreßbestimmung* vorgenommen werden. Adressen sind (von Sonderfällen abgesehen) nicht artspezifisch (siehe *Typisierung*). Maschinenadressen (im allgemeinen) bilden eine linear geordnete, beschränkte Menge, die ordnungsisomorph zu den natürlichen Zahlen ist.

Adressen werden bei der Programmierung in symbolischer oder *numerischer* Form angegeben. Symbolische Adressen wandelt ein *Interpre-*

tierer, Assemblierer oder Übersetzer in numerische Adressen um.
Hoffmann, H.-J.

Adresse von Objekten in Rechnernetzen

Teilgebiet: Rechnernetze und verteilte Systeme

Im *Rechnernetz* eine Folge von Zeichen zur netzeindeutigen Kennzeichnung (Lokalisierung) von Objekten, z.B. des Absenders bzw. *Empfängers* einer *Nachricht*. Der Namens- (\rightarrow *Bezeichner*) bzw. *Adreßraum* wird üblicherweise hierarchisch vergeben. In einem Kommunikationssystem (\rightarrow *Kommunikationssystem*) können mögliche Komponenten einer Nachrichtenadresse sein:

- Kennzeichen für das Rechnernetz, in dem sich der Adressat befindet,
- innerhalb eines Netzes eindeutiges Kennzeichen für den *Kommunikationsrechner*,
- innerhalb eines Kommunikationsrechners eindeutiges Kennzeichen für den *Arbeitsrechner* bzw. das Netzwerksteuerprogramm im Arbeitsrechner,
- innerhalb eines Arbeitsrechners eindeutiges Kennzeichen des Benutzers bzw. seines Prozesses,
- für einen Prozeß eindeutiges Kennzeichen des *Prozeßeingangs/Prozeßausgangs* (*port*).
Hegering

Adressenfortschaltung, lineare

Teilgebiet: Programmierung
recursive address calculation

Die lineare Adressenfortschaltung ist ein Hilfsmittel zur Beschleunigung von *Schleifen*, in deren Laufbereich *Zugriffe* auf Elemente eines Feldes (\rightarrow *Feld, mehrdimensionales*) auftreten. Während nach nichtoptimierender Übersetzung bei jedem Feldzugriff eine erneute Auswertung der *Speicherabbildungsfunktion* erfolgt, nutzt eine optimierende Übersetzung mit linearer Adressenfortschaltung die Tatsache aus, daß die Speicherabbildungsfunktion linear von den Feldindizes abhängt: bei konstanter Schrittweite genügt es, statt erneuter Auswertung der Speicherabbildungsfunktion die Adresse dieses Feldelements aus dem vorherigen Schleifendurchlauf um einen konstanten Betrag zu erhöhen oder zu erniedrigen. Diese Differenz hängt nur von der Schrittweite der *Laufvariablen* und den Angaben in der zugehörigen Feldvereinbarung ab. Götter

Adressenrechnung \rightarrow *Adreßbestimmung*

Adressenregister \rightarrow *Adreßregister*

Adressenteil \rightarrow *Adreßteil*

Adressenversatz

Teilgebiet: Programmierung

displacement

Synonyme: Distanzadresse; Zeilennummer
Siehe auch: Adressenfortschaltung, lineare

Die (*numerische*) *Distanz* zwischen der Adresse eines anzusprechenden Speicherplatzes (auch *Relativadresse* genannt) und der Adresse eines als Bezugspunkt verwendeten Speicherplatzes (Programmangfang, Anfang eines Datenbereichs u.ä.), der *Basisadresse*. In einer Adressenangabe bildet der Adressenversatz (im allgemeinen) zusammen mit der Adresse des *Registers*, das die Basisadresse aufgenommen hat, eine *zusammengesetzte Adresse*, die noch der *Adreßbestimmung* zu unterwerfen ist. Bei der Verwendung *indizierter Adressen* auch die Distanz zwischen in aufeinanderfolgenden *Schleifen*durchläufen angesprochenen Speicherplätzen.
Hoffmann, H.-J.

Adreßformat

Teilgebiet: Programmierung
address format

Synonym: Befehlstyp

Es hält die Zahl und die Bedeutung der Adressen eines Befehls fest (siehe *Befehlsformat*). Von Wichtigkeit sind besonders 0-Adreßbefehle, 1-Adreßbefehle (\rightarrow *Eins-Adreßbefehl*), 2-Adreßbefehle (\rightarrow *Zwei-Adreßbefehl*) und 3-Adreßbefehle (\rightarrow *Drei-Adreßbefehl*). Die Zahl entspricht der Anzahl von (Operanden-) Adressen im Befehl. Wegen der meist zweistelligen *Operationen*, die durch Befehle realisiert werden, wären 3-Adreßbefehle am naheliegendsten (Adressen der beiden *Operanden* und zur Ablage des Resultats), man findet aber überwiegend eine geringere Adressenzahl (kürzere *Befehlswoorte*). Bei 2-Adreßbefehlen nimmt der Speicherplatz eines beteiligten Operanden auch das *Resultat* auf. Bei 0- und 1-Adreßbefehlen werden beide oder einer der Operanden einem (festgelegten oder) als *Kellerspeicher* organisierten *Speicherbereich* entnommen bzw. ein Resultat dort abgelegt; beteiligt ist dann ein Keller(-pegel)-register, dessen Wert bei der *Befehlsausführung* selbsttätig mitgeführt wird, so daß Operanden (z.B. zwei zurück) bzw. das Resultat (eins höher) immer richtig erreicht werden. Erfolgt die Adreßfortschaltung nicht durch Aufaddieren von Eins zur Befehlsadresse, sondern ist die Adresse des nächsten Befehls (*Folgeadresse*) auch jeweils im Befehl enthalten, spricht man von (n+1)-Adreßformat, $n \in \{0,1,2\}$.
Hoffmann, H.-J.

Adreßfortschaltung \rightarrow *Adreßformat; Adressenfortschaltung, lineare*

adressierbare Position \rightarrow *Punkt, adressierbarer*

adressierbarer Punkt \rightarrow *Punkt, adressierbarer*

adressierbarer Speicher → *Zugriff, indirekter*

Adressieren

Teilgebiet: Programmierung
 Siehe auch: Adressierung

Der Vorgang der *Adreßbestimmung*.

Hoffmann, H.-J.

Adressierung

Teilgebiete: Programmierung; Rechnerorganisation
addressing

Zuordnung von Adressen zu Daten oder Befehlen zur Lokalisierung der Speicherplätze, in denen sie untergebracht sind.

Flik

Adressierung, direkte → *Adresse, direkte; Hash-Algorithmus*

Adressierung, indirekte → *Adresse, indirekte; Hash-Algorithmus*

Adressierung, segmentierte

Teilgebiet: Programmierung
segmented addressing

Verfahren, einen *Adreßraum* der Breite 2^p mit *Registern* der Stellenzahl $q < p$ zu *adressieren*. 2^q zusammenhängende Speicherplätze bilden ein Segment; hierin erlaubt ein (1.) Register mit seinem Inhalt d relativ zu adressieren. Segmente *überlagern* sich: Eine *Speicherzelle* liegt beispielsweise im Segment i_1 und habe darin eine *Relativadresse* d_1 ; im darauffolgenden Segment $i_2 = i_1 + 1$ hat sie dann bei Segmentabstand 2^{p-q} die Relativadresse $d_2 = d_1 - 2^{p-q}$. Die Anfangsadresse (→ *Basisadresse*) des Segments als *Basisadresse* liefert ein weiteres (2.) Register; dessen Inhalt wird zur *Adreßbestimmung* mit 2^{p-q} multipliziert und zu d hinzuaddiert.

Dieses Verfahren wird mit $p = 20$ (2^p Byte = 1 MByte) und $q = 16$ (2^q Byte = 64 kByte), d.h. mit $p - q = 4$ und Segmentabstand $2^4 = 16$, bei 80x86 Prozessoren (im sog. real-Mode) angewandt; spezielle Register DS für Datensegment, CS für Codesegment (mit einem 16 Bit breiten Befehlszählregister (→ *Befehlszähler*) für *Distanz*), SS für Kellersegment und ES für spezielle Zwecke liefern die jeweilige *Segmentanfangsadresse*.

Wenn sich (bei 80x86-Prozessoren im real-Mode) Daten-, Code- und Kellersegment an der gleichen Stelle befinden, spricht man von *tiny-Modell* der Segmentaufteilung; es wird bei ausführbaren Programmen mit *Datei* suffix „.COM“ eingesetzt.

Hoffmann, H.-J.

Adressierung, virtuelle

Teilgebiete: Programmierung; Rechnerorganisation

virtual addressing, virtual address

Soll für ein Programm transparent sein, welche *Maschinenadresse* ein Speicherplatz (→ *Speicherzelle*) hat, der bei der *Programmausführung* angesprochen wird, oder ob der Speicherplatz sich überhaupt im Arbeitsspeicher (→ *Hauptspeicher*) und nicht ausgelagert auf einem *Hintergrundspeicher* befindet, spricht man von einem *virtuellen Speicher*. Er wird entweder durch *Paging* oder durch *Swapping* realisiert. Der *Adreßraum* eines virtuellen Speichers kann kleiner (z.B. bei *Swapping*), gleich oder größer als der des tatsächlich verwendeten Arbeitsspeichers sein; er ist in Seiten (oder Segmenten) organisiert. Bei *Swapping* verwendet man (meist) selbstverschiebliche Programmbereiche, so daß keine besonderen Maßnahmen bei der *Adreßbestimmung* erforderlich werden. Bei *Paging* treten virtuelle Adressen auf (d.h. *Maschinenadressen* im *virtuellen Adreßraum*); Adressen (typischerweise die Seitennummer) ersetzt durch Anfangsadressen von *Kacheln*, so daß (wahre, tatsächliche) *Maschinencodadressen* (→ *Maschinsprache*) herauskommen.

Virtuelle Adressierung ist Grundlage von Teilnehmerrechnersystemen.

Hoffmann, H.-J.

Adressierungseinrichtung → *DMA-Controller*

Adreßkette → *Stücklistenprozessor*

Adreßkonstante

Teilgebiet: Programmierung
address constant
 Synonym: ADCON

Eine *numerische Adresse*, die zum Zeitpunkt des Ladens als *Maschinenadresse* in einem (für die Aufnahme einer Zahl aus dem *Adreßraum* geeigneten) Wort des Speichers zum Zeitpunkt des Ladens steht und, sobald sie in ein *Register* übertragen ist, als *Basisadresse* für einen Programm- oder Datenbereich dienen kann. Ist ein solcher Bereich also im Speicher zu verlegen, genügt es, den Wert der *Adreßkonstanten* vor einem Ladevorgang (→ *Laden und Ausführen*) entsprechend zu setzen. Sind Programmbereiche selbstverschieblich, entfällt die Notwendigkeit, auf eine *Adreßkonstante* zurückzugreifen.

Hoffmann, H.-J.

Adreßliteral → *Operand an Adreßposition*

Adreßmodifikation

Teilgebiet: Programmierung
 Siehe auch: Adreßbestimmung

Steht die Adresse eines *Operanden* erst zur *Laufzeit* fest oder soll ein Befehl, z.B. in einer *Schleife*, auf verschiedenen Operanden arbeiten, so kann die *Adreßmodifikation* zur Bestimmung der

Adreßraum

Operandenadresse verwendet werden. Dabei wird die aktuelle Adresse durch Addition eines konstanten *Adreßteils* und eines *dynamisch* bestimmbaren Teils bei *Befehlsausführung* errechnet. Der dynamische Teil muß in einem *Register* gespeichert werden. Würges

Adreßraum

Teilgebiet: Programmierung
address space; name space

Der durch *numerische Adressen* ausschöpfbare Bereich natürlicher Zahlen, meist von Null bis zu einer Zahl, die um Eins geringer als eine Zweierpotenz ist. Vor der *Adreßbestimmung* liegt u.U. ein *virtueller Adreßraum* vor, danach ein *Adreßraum*, dessen Größe durch die Größe des Speichers bestimmt ist. Man spricht von regionalem *Adressieren* oder Segmentadressierung (→ *Segment¹*), wenn mehrere Adreßräume zur Verfügung stehen. Hoffmann, H.-J.

Adreßraumhierarchie → *Namensraum*

Adreßrechenwerk → *Basisadresse*

Adreßrechnung → *Adreßbestimmung*

Adreßregister

Teilgebiet: Rechnerorganisation
address register

Ein *Register* zur Aufnahme einer Adresse, z.B. zur *Adressierung* eines Speichers (Speicheradreßregister). Flik

Adreßsignal → *Adreßbus*

Adreßsymbol → *Adresse, symbolische*

Adreßteil

Teilgebiet: Programmierung
address part

Synonym: Adressenteil

Siehe auch: Adreßformat

Im *Befehlsformat* einer Maschinen- oder *Assemblierersprache* vorgesehener Bestandteil zur Aufnahme der Angaben für eine (zusammengesetzte) Adresse. Im *Befehlsformat* können mehrere Adreßteile vorgesehen sein. Hoffmann, H.-J.

Adreßübersetzung

Teilgebiet: Rechnerorganisation
address translation

Umsetzung virtueller Adressen in reale Primärspeicheradressen. Die Adreßübertragung wird von der Zentraleinheit im Zuge der *Befehlsausführung* automatisch vorgenommen. Dazu werden vom *Betriebssystem* im Primärspeicher (→ *Hauptspeicher*) aufgebaute Übersetzungstabellen herangezogen, welche die Abbildung vir-

tueller Adressen auf reale Primärspeicheradressen beschreiben. Durch spezielle *Hardware* zur automatischen Speicherung häufig benutzter Paare *virtuelle*-reale Adresse werden Primärspeicherzugriffe auf Übersetzungstabellen (meist mehrere pro Übersetzungsvorgang) vermieden; dadurch wird die Adreßübersetzung verkürzt. Müller-Stoy

Adreßumsetzung → *Adreßbestimmung*

Adreßverkettung → *Teilestammsatz*

Adreßwiederholung

Teilgebiet: Programmierung
repetitive addressing

Entnahme einer Angabe zur *Adreßbestimmung*, insbesondere bei *zusammengesetzten Adressen*, nicht aus dem oder einem *Adreßteil* des zur Ausführung anstehenden Befehls, sondern aus dem *Adreßteil* eines früher ausgeführten Befehls. Erlaubt, falls in einem Programmstück nur ein kleiner Ausschnitt aus einem *Adreßraum* wiederholt angesprochen wird, sehr klein gehaltene Adreßteile im *Befehlsformat* (in den *Leitwerken* einiger Rechenanlagen, insbesondere bei Kleinrechnern (→ *Minicomputer*) und Kleinrechnern (→ *Microcomputer*), möglich). Hoffmann, H.-J.

Adreßzähler

Teilgebiet: Rechnerperipherie

Ein Adreßzähler ist ein zählendes *Register*, dessen Inhalt zur *Adressierung* eines Speichers verwendet wird. Seine Aufgabe ist es, eine *Sequenz* von Informationseinheiten (*Bytes* oder Worte), die in konsekutiven *Speicherzellen* lokalisiert sind, für schreibenden oder lesenden Zugriff auszuwählen. Beispiele für die Verwendung von Adreßzählern sind: *Befehlszähler*, zählende *Pointerregister* und Adreßzähler in *DMA*-Bausteinen. Flik

ADT → *Datentyp, abstrakter*

ADV → *Datenverarbeitung (im weiteren Sinne)*

ADV-Fachmann → *Datenverarbeitungsspezialist*

ADV-Normen → *ADV-Recht*

ADV-Organisation → *Informationsorganisation*

ADV-Organisationsrecht

Teilgebiet: Informatik in Recht und öffentlicher Verwaltung

ADP organisation statutes

Die Menge der (materiellen oder formellen) Gesetze und Rechtsverordnungen, die vor allem die

Informationsorganisation und ihr Verhältnis zur herkömmlichen Verwaltungsorganisation mittels einer Reihe von spezialisierten Behörden und Koordinierungsorganen sowie informationstechnischen Vorschriften ordnen.

Erläuterung:

Im Zuge der *Verwaltungsautomation*, vor allem der öffentlichen Hand, entstanden aus technischen, finanziellen und organisatorischen Gründen (vor allem anlässlich der Schaffung integrierter Informationssysteme) eine, die herkömmliche Organisation der Verwaltung, durchdringende neue Organisation der *Informationsverarbeitungsprozesse* (*Rechenzentren*, in Gestalt regionaler oder kommunaler *Datenzentralen*, Koordinationsstellen, EDV-Ausschüsse), die Informationsorganisation, teils neben der herkömmlichen Verwaltung (früher sog. „Dissoziation der Verwaltung“, heute auch im Rahmen des *outsourcing*), teils bis zur Ununterscheidbarkeit miteinander verflochten (als „Büroautomation“, heute zunehmend telekommunikativ vernetzt).

Deren Regelung ist Aufgabe der ADV-Organisationsgesetze von Bund und Ländern, ergänzt durch das Telekommunikations- und Datenschutzrecht (→ *Datenschutzvorschriften*).

Steinmüller

ADV-Recht

Teilgebiet: Informatik in Recht und öffentlicher Verwaltung

ADP statutes

Synonyme: ADV-Normen; ADV-Vorschriften; EDV-Gesetze; EDV-Vorschriften

Menge der Rechtsnormen (d.h. Gesetze im materiellen oder im formellen Sinn), die Prozesse oder Institutionen der automatisierten Datenverarbeitung einschließlich Telekommunikation regeln. Erläuterung:

Vorschriften, die aussagen, daß ein bestimmter automatisierter (→ *Automatisierung*) Informationsprozeß geboten, gestattet oder verboten ist (z.B. Anzeige- und Meldepflichten, *Datenbankkonzession*, berufliche Schweigepflichten, Zeugnisverweigerungsrechte). Sie haben den automatisierten Informations- einschl. Kommunikationsprozeß unmittelbar zum Gegenstand der Regelung; ihr Inhalt ist es, *computergestützte* Informations- und Kommunikationsprozesse zu ermöglichen, durchzusetzen oder zu unterbinden. Sie definieren, welche Informationen bzw. Daten in welchem *Kanal* mit oder ohne welche technischen Geräte (→ *Peripheriegerät*) und/oder Vorkehrungen fließen müssen, können oder (nicht) dürfen.

Kontext:

Die umfassenden Möglichkeiten des ADV-Einsatzes, seine gesellschaftlichen und politischen Auswirkungen bedürfen der Steuerung durch den Gesetzgeber, um den ADV-Einsatz einschließlich der Telekommunikation rechts- und sozialverträglich zu gestalten, aber auch Risiken zu erkennen und Fehlentwicklungen zu vermeiden. ADV-Rechtsnormen dienen der Aufgabe, solchen unerwünschten Fehlentwicklungen vorzubeugen, ggf. unerwünschte Folgen zu beseitigen oder positive Ziele vorzulegen.

Arten:

- *ADV-Organisationsrecht*
- Datenverarbeitungsgesetze (z.B. BundesmeldeG, BundeszentralregisterG, DEVO = Datenerfassungsverordnung; DÜVO = Datenübermittlungsverordnung)
- Datenzugangsgesetze (z.B. U.S. Freedom of Information Act) und Datenschutzgesetze (z.B. *Bundesdatenschutzgesetz* = BDSG; *Länderdatenschutzgesetze*; „bereichsspezifische Regelungen“; wie § 35 SGB-AT, § 30 BVerwVfG)
- Datenübertragungsrecht, also das Telekommunikationsrecht (→ *Telekommunikation*¹)
- sonstige informationstechnikspezifische Gesetze; z.B. früher *Bildschirmtext* erprobungsgesetze; ergonomische Vorschriften.

Zunehmende Bedeutung erhält das EG-Recht zur ADV (→ *Datenverarbeitung* (im weiteren Sinne)) und Telekommunikation.

Steinmüller

ADV-Recht (gelegentlich) → *Informationsrecht*

ADV-Vorschriften → *ADV-Recht*

AD-Wandler → *Analog-Digital-Umsetzer*

Ähnlichkeitsfunktion

Teilgebiet: Linguistische Datenverarbeitung
similarity function

Eine Ähnlichkeitsfunktion ist auf der Menge der Paare einer Grundmenge definiert und ergibt für jedes Paar einen nichtnegativen Zahlenwert (Ähnlichkeit). Die Elemente der Grundmenge sind als Zahlenvektoren, sogenannte Merkmalsvektoren, dargestellt, die durch die Ähnlichkeitsfunktion so verknüpft werden, daß sich ein Maß für die *Übereinstimmung* ergibt. Ihr Maximum erreicht die Ähnlichkeitsfunktion für zwei gleiche Merkmalsvektoren. Zweck der Berechnung der Ähnlichkeitsfunktion ist die *Klassifizierung*. Meistens wird für diesen Zweck die Distanzfunktion berechnet.

Frenkel

Ähnlichkeitskoeffizient → *Ähnlichkeitsmaß*

Ähnlichkeitsmaß

Teilgebiet: Information Retrieval

Ähnlichkeitsmatrix

similarity measure

Synonym: Ähnlichkeitskoeffizient

Ein Ähnlichkeitsmaß ordnet einem Paar von Objekten (O_i, O_j) eine reelle Zahl S_{ij} zu, die um so größer ist, je ähnlicher die Objekte zueinander sind. Dabei fordert man

a) $0 \leq S_{ij} \leq 1$

b) $S_{ij} = S_{ji}$

c) $S_{ii} = 1$

Grundlage vieler Ähnlichkeitsmaße ist die *Vierfeldertafel*. Dabei wird vorausgesetzt, daß die Objekte durch *binäre* Merkmale (\rightarrow *Attribut*) beschrieben werden. Verschiedene Ähnlichkeitsmaße entsprechen verschiedenen Vorstellungen von Ähnlichkeit. Welches Maß gewählt wird, hängt vom Anwendungsgebiet ab. Wichtige Maße, die bei experimentellen *Dokumenten-Retrieval-Systemen* benutzt werden, sind das *Cosinusmaß*, der *Koeffizient von Jaccard* sowie das *Überlappungsmaß*.

Bollmann

Ähnlichkeitsmatrix

Teilgebiet: Information Retrieval

similarity matrix

Synonym: Matrix paarweiser Ähnlichkeiten

Die Ähnlichkeitsmatrix

$$(S_{ij}) \quad i=1, \dots, N, j=1, \dots, N$$

gibt die Ähnlichkeit S_{ij} zwischen dem i -ten und j -ten Objekt einer endlichen Objektmenge an. Die Zahlen S_{ij} werden mit *Ähnlichkeitsmaßen* berechnet. Die Ähnlichkeitsmatrix ist Grundlage vieler *Klassifikationsverfahren*.

Bollmann

AEK

Teilgebiet: Informatik im Bankwesen

Abkürzung von: Anschalteinheit für Kartentelefone

Eine AEK steuert bis zu 172 Kartentelefone. Die AEK enthält ein Sicherheitsmodul zur Abwicklung der Sicherheitsverfahren mit den *Chipkarten* u.a. auch zur Verwaltung von Sperrlisten.

Martiny

Änderung einer Datenbank \rightarrow *Datenbank, Änderung einer*

Änderungen, Freigeben von

Teilgebiet: Nebenläufige Prozesse
commit

Sichtbarmachen der von einer *Transaktion* durchgeführten Änderungen für andere *Transaktionen*. Beim *Zweiphasen-Sperrprotokoll* geschieht dies durch Freigabe der Objektsperren.

Dadam

Änderungsprotokollierung \rightarrow *Logging*

Äquivalenz

Teilgebiet: Schaltwerke und Schaltnetze

Eine zweistellige *boolesche Funktion* mit der Funktionstabelle

a	b	$a \equiv b$
0	0	1
0	1	0
1	0	0
1	1	1

Als Funktionssymbole sind gebräuchlich: $a \sim b$,
 $a \leftrightarrow b$ *Reusch*

Äquivalenz, aussagenlogische \rightarrow *Bisubjunktion*

Äquivalenz, logische

Teilgebiet: Logik

Siehe auch: Interpretation in der Logik

Zwei Formeln (Aussagen) sind *logisch äquivalent*, wenn sie bei jeder Interpretation und *Belegung* gleiche *Wahrheitswerte* haben. Gleichwertig ist: Zwei Formeln A, B sind logisch äquivalent, wenn die zugehörige *Bisubjunktion* $A \leftrightarrow B$ allgemeingültig ist.

Müller

Äquivalenzklasse

Teilgebiet: Grundlegende Definitionen
equivalence class

Ist R eine *Äquivalenzrelation* auf einer Menge A , so heißt die Menge $M = \{b \mid b \in A, (a,b) \in R\}$ der zu einem Element $a \in A$ äquivalenten Elemente $b \in A$ eine *Äquivalenzklasse* zu R . Die verschiedenen in A vorhandenen Äquivalenzklassen M bilden eine *Partition* von A . Umgekehrt induziert jede *Partition* $P: A = M_1 + M_2 + \dots$ eine *Äquivalenzrelation* R auf A : $(a,b) \in R$ genau dann, wenn a und b in derselben Menge (Klasse) M_j von P liegen.

Bock

Äquivalenzrelation

Teilgebiet: Grundlegende Definitionen
equivalence relation

Eine (*binäre*) *Relation* $R, R \subset A \times A$ auf einer Menge A heißt *Äquivalenzrelation*, falls R die folgenden drei Bedingungen erfüllt:

- (1) für jedes $a \in A$ gilt $(a,a) \in R$ (*Reflexivität*),
- (2) ist $a \in A$ und $b \in A$ und gilt $(a,b) \in R$, dann gilt auch $(b,a) \in R$ (*Symmetrie*),
- (3) sind $a,b,c \in A$ und gilt $(a,b) \in R$ und $(b,c) \in R$, dann gilt auch $(a,c) \in R$ (*Transitivität*).

Czap

Äquivalenz von Grammatiken

Teilgebiet: Linguistische Datenverarbeitung

Man unterscheidet zwischen schwacher und starker *Äquivalenz* von Grammatiken:

- (i) Zwei *Grammatiken* sind schwach äquivalent, wenn sie dieselbe Sprache festlegen.
- (ii) Zwei *Grammatiken* sind stark äquivalent, wenn sie außerdem dieselben *Zerlegungen der Zeichenketten* vornehmen, oder m.a.W. den sprachlichen Einheiten dieselbe *Strukturbeschreibung* zuordnen.

Von Bedeutung ist die schwache *Äquivalenz* für die Konstruktion von sog. *Normalform-Grammatiken*, die für die maschinelle Abarbeitung von Vorteil sein können. Hellwig

Ästhetikprogramm

Teilgebiet: Desktop Publishing
aesthetic program

Programmgesteuerte Routinen, die beim rechnergesteuerten (→ *computergestützt*) Satz dafür sorgen, daß Buchstabenabstände bei bestimmten Groß- und Kleinbuchstabenkombinationen so variiert werden, daß ein ästhetisch angenehmer, gleichmäßiger Gesamteindruck entsteht. Auch Wortzwischenräume, besonders in Verbindung mit Satzzeichen und *Versalien*, werden durch geeignete Routinen ausgeglichen. Ehlers

after image

Teilgebiet: Grundlegende Definitionen
 Kopie eines Datenbanksatzes bzw. einer *Tabelle* nach ihrer Änderung auf ein *Log*.
Hammann-Kloss

after-image → *Logging*

Agent¹

Teilgebiet: Programmierungsmethodik
 Synonym: Akteur
 Agenten nennt man die selbständig und miteinander interagierenden (*Software-* oder *Hardware-*) Bestandteile eines verteilten Systems. Partsch

Agent²

Teilgebiet: Telekommunikation
 Mitarbeiter/Sachbearbeiter in einem *Communications-* bzw. *Call Center*. Kuhn

Aggregation

Teilgebiet: Programmierungstechnik
aggregation
 Im *Entity-Relationship-Datenmodell* die Betrachtung eines Beziehungstyps als Objekttyp oder als Zusammensetzung von Objekten und Beziehungen. Schneider

Aggregation

Teilgebiet: Grundlagen von Informationssystemen
 Siehe auch: *Aggregation*

Zusammenfassung von Dateneinheiten, denen ein Zusammenhang unterstellt wird. Dieser Zusammenhang wird dann genutzt, um besonders leicht von einer Dateneinheit zu einer weiteren zu gelangen. Basiert der Zusammenhang speziell auf der *Bestandteils-(part-of)-Relation*, so wird die Zusammenfassung als auch neue Dateneinheit in ihrer Gesamtheit manipuliert. Häufig wird die Aggregation auch als Hinweis auf eine zusammenhängende Ablage auf dem *Hintergrundspeicher* gedeutet. Lockemann

AI → *Artificial intelligence*

Aiken-Code

Teilgebiet: Schaltwerke und Schaltnetze
 Der Aiken-Code ist ein vierstelliger *Binär*code zur *Darstellung* von *Dezimalziffern* (→ *Dezimalsystem*). Die *Stellenwertigkeit* ist 2-4-2-1. Entsprechend wird dieser Code nach *DIN 44800* 2-4-2-1-Code genannt. Die *Dezimalziffern* 0 bis 4 sind mit *Tetraden* in der entsprechenden *dualen Darstellung (Dualsystem)* verschlüsselt (→ *Ver-/Entschlüsselung¹*), die *Dezimalziffern* 5 bis 9 jedoch mit *Tetraden*, die der dualen Darstellung der *Dezimalzahlen* 11 bis 15 entsprechen. Durch diese Art der *Codierung* wird das Bilden des *Neuner-Komplements* vereinfacht. Mittels *Neuner-Komplement* läßt sich eine *Subtraktion* sehr einfach auf eine *Addition* zurückführen. Der Aiken-Code hat ähnliche Vorteile wie der *Drei-Exzeß-Code* (→ *Exzeß-3-Code*), bietet aber nicht wie letzterer die Möglichkeit, bestimmte *Maschinenfehler* zu erkennen. Dafür kann jedoch jeder *Stel-*

	Dezimalziffer	Aiken	Code	
	0	0 0	0 0	
	1	0 0	0 L	
	2	0 0	L 0	
	3	0 0	L L	
	4	0 L	0 0	
Pseudo-tetraden	–	0 L	0 L	L
	–	0 L	L 0	L
	–	0 L	L L	L
	–	L 0	0 0	0
	–	L 0	0 L	L
	–	L 0	L 0	0
	5	L 0	L L	
	6	L L	0 0	
	7	L L	0 L	
	8	L L	L 0	
	9	L L	L L	
	Stellenwert	2 4	2 1	

Tabelle: Aiken-Code

le im Binärsystem ein bestimmter Stellenvorrat zugeordnet werden. Nilsson

Aiken, Howard

Teilgebiet: Geschichte der Informatik

Erbaute an der Universität Harvard mit Unterstützung der Firma IBM 1944 die elektromechanische Rechenanlage Mark I. Die Anlage besitzt bedingte *Sprungbefehle*. Knödel

AIM → *Avalanched Induced Migration*

AIV → *Datenverarbeitung (im weiteren Sinne)*

Akkreditierung

Teilgebiet: Qualitätsmanagement

Verfahren, in dem eine maßgebliche Stelle formell anerkennt, daß eine Stelle oder Person kompetent ist, bestimmte Aufgaben auszuführen (nach *DIN EN 45 020*, Ausgabe April 1994). Möller

Akkumulator

Teilgebiet: Rechnerorganisation

Siehe auch: Einadreßrechner

Register, das für Rechenoperationen (→ *Rechenprozeß*) benutzt wird, wobei es ursprünglich einen *Operanden* und nach durchgeführter *Operation* das Ergebnis enthält.

Anm.: Da ein Akkumulator sowohl *Quelle 1* als auch Ziel einer Operation ist, kann der zugehörige Befehl

Akku ← Akku op Quelle 2
mit einer Adresse (der von Quelle 2) auskommen. Klar

Akkumulatormaschine → *Einadreßrechner*

Akteur → *Agent*¹

Aktion

Teilgebiet: Informatik in der Betriebswirtschaft
action

Synonyme: Handlung; Tätigkeit

Siehe auch: *Prozeß*²

Unter Aktion wird ein elementarer (einstufiger und damit *sequentieller*) Teilprozeß verstanden. Ein *Parallelismus* ist also nicht mehr möglich.

Unter betriebswirtschaftlich-organisatorischer Sicht wird darunter ein versachlichter Komplex von Verrichtungen verstanden, die von einer Stelle ausgeführt werden. Der Verrichtungskomplex, der Teil der Gesamtaufgabe ist, wird dabei durch synthetische Zusammenfassung analytisch gewonnener Teilaufgaben gebildet. Die Verbindung des Begriffs Aktion und Stelle führt zum *Begriff* der Aktionseinheit. Sie umfaßt damit in Abhängigkeit von dem *Prozeß*ausschnitt, den sie

auszuführen hat, einen oder mehrere *Aktionsträger*. BIFOA

Aktionsfehler

Teilgebiet: Interaktive Systeme, Mensch-Maschine-Dialog

action error

Fehlerkategorie (→ *Fehlzustand*) im Rahmen der *Nutzung* interaktiver (→ *Interaktion*) Systeme. Ein *Aktionsfehler* tritt auf, wenn aufgrund von Unkenntnis oder Vergessen eine notwendige *Aktivität* vom Benutzer des Systems nicht ausgeführt wird und dadurch ein Fehler produziert wird. Freiburg

Aktionsregel

Teilgebiete: Kognitive Methoden – Künstliche Intelligenz; Wissensverarbeitung, Expertensysteme

action rule

Aktionsregeln bilden den Kontrollmechanismus für die *Inferenzmaschine* in *wissensbasierten* Systemen (*Expertensystemen*). Jede *Aktionsregel* definiert eine *Bedingung*, verbunden mit einer *Aktion*, die bei Zutreffen der *Bedingung* ausgeführt wird. *Aktionsregeln* arbeiten nach dem Prinzip des *Forward Chaining*. Freiburg

Aktives Sehen

Teilgebiet: Kognitive Methoden – Künstliche Intelligenz

active vision

In der automatischen *Bildanalyse* (→ *Bildverarbeitung*) der Übergang von *Verfahren*, die eine vorgegebene *Bildfolge* interpretieren, zu *Verfahren*, welche „aktiv“ die zur Lösung einer gegebenen Aufgabe geeigneten und erforderlichen *Bilder* bzw. *Bildausschnitte* ermitteln, aufnehmen und auswerten und für die Aufgabe *Irrelevantes* ignorieren.

Aktives Sehen in diesem Sinne erfordert die algorithmische Bestimmung der für die Aufgabe interessanten *Bildbereiche* bzw. *Objekte* und *Ergebnisse*. Dieses geschieht zunächst durch relativ einfach zu berechnende *Auffälligkeitsmaße*, die zum Beispiel *Bewegungen*, bestimmte *Farben*, *Texturen* oder *Größen* detektieren. Unter *Programmkontrolle* werden die auffälligen Bereiche fokussiert, im *Ausschnitt* vergrößert aufgenommen, durch *aktive* und *autonome* Prozesse über die *Zeit* verfolgt und bei Bedarf *modellgestützt* interpretiert. Abhängig von den *Ergebnissen* der *Verarbeitung* wird die weitere Strategie zur *Aufnahme* und *Verarbeitung* von *Bildern* bestimmt. Das Vorbild ist der menschliche *Beobachter*, der seine *Umgebung* aktiv *exploriert* und in ihr *zielgerichtet* agiert. Im allgemeinen werden beim *ak-*

tiven Sehen also Sensoren, *Aktoren* und internes *Wissen* (Modelle) in einer geschlossenen Interaktionsschleife von Sensorik und Aktorik eingesetzt.

Das Ziel ist die effiziente Bewältigung eines definierten *Problems* wie kollisionsfreie Navigation in einem Raum oder Ergreifen eines bestimmten Objekts, aber nicht vorrangig die (vollständige) symbolische Beschreibung von Szenen. Das Gebiet ist noch stark in der Entwicklung begriffen. Niemann

Aktivierungskontour → *Kontour*

Aktivität bei Leistungsmessung

Teilgebiet: Messung, Modellierung und Bewertung von Rechensystemen

In der Leistungsbewertung (→ *Leistungsermittlung*) von Rechensystemen (→ *Datenverarbeitungssystem*) bezeichnet man einen interessierenden *Hardwarezustand* oder den Ablauf eines Programmabschnitts als Aktivität. Aktivitäten haben eine Dauer oder *Laufzeit*. Meßtechnisch sind Aktivitäten in der Regel durch ein Anfangs- und Endereignis gekennzeichnet (siehe: Ereignis). Deren *Zeitstempeldifferenz* stellt die Aktivitätsdauer dar. Klar

Aktor → *Datenverarbeitungsaufgabe*

aktueller Parameter → *Parameterbehandlung*

Akustikkoppler

Teilgebiet: Rechnerperipherie
acoustic coupler

Ein Gerät, das ähnlich einem *Modem* der *Datenübertragung* dient. Die *analoge* Verbindung mittels Fernsprecher erfolgt durch Auflegen des Telefonhörers auf eine Gummiauflage. Ehlers

Akzeptanzforschung

Teilgebiete: Gesellschaftsrelevante Aspekte der Informatik; Informatik in der Betriebswirtschaft
acceptance research

Untersuchung zur Verbreitung und Übernahme von Innovationen (z.B. neue *Produkte*, Methoden, Ideen) im Verlauf von Diffusionsprozessen. Zur Erklärung und Vorhersage der Verbreitung neuer Informationstechnik werden Schätzungen des Markt- bzw. Nutzerpotentials und des zeitlichen Verlaufs der Übernahme durch einzelne Adoptergruppen vorgenommen. Da Akzeptanz (→ *Akzeptanzproblem*) kein operationaler Begriff im Rahmen einer erklärungsstarken Diffusionstheorie ist, finden in Untersuchungen sehr unterschiedliche Prädiktorvariablen Anwendung, so z.B. Einstellungen und Interessen, Nutzungs- und Käuferabsichten, erwartete Vor- und Nach-

teile, Entwicklung von Ausgabebereitschaft und Kaufkraft einzelner Bevölkerungsgruppen. Für Absatzprognosen, sog. Begleitforschung sowie Technikfolgeschätzungen lassen sich Methoden der empirischen Sozialforschung und der Marktforschung anwenden, so z.B. kontrollierte Feldexperimente mit Vor- und Nacherhebungen sowie Variation der unabhängigen Größen (z.B. der Preise, *Betriebskosten*, technischen Konzeption). Kuhlmann

Akzeptanzproblem

Teilgebiet: Informatik in der Betriebswirtschaft
problem of acceptance

Synonyme: mangelnde Nutzung; fehlende Nutzungsbereitschaft oder Nutzungsfähigkeit

Computergestützte *Informationssysteme* – insbesondere planungsorientierte „*Managementinformationssysteme*“ (MIS), aber auch Arbeitsplatz-Computer (APC) – werden häufig von ihren *Anwendern* in Fachabteilungen (Benutzern) nicht oder nur unzureichend genutzt. Diese fehlende Akzeptanz kann vielfach auf

- sachliche/aufgabenlogische wie auch
- individuelle/soziale

Faktoren zurückgeführt werden.

Wird ein Informationssystem zunächst systemtechnisch (→ *System'*) entwickelt und implementiert, anschließend erst (durch *Schulung* etc.) organisatorisch implementiert, so besteht die Gefahr, daß wesentliche

- sachliche *Anforderungen*
- Interessen und Bedürfnisse

direkt oder indirekt Betroffener nicht berücksichtigt werden und deshalb das System nicht angenommen, evtl. sogar sabotiert wird. Aus diesem Grunde sollte angestrebt werden, die von der Systementwicklung Betroffenen angemessen – nach dem Grad ihrer Betroffenheit – am Gestaltungsprozeß zu beteiligen. BIFOA

Akzeptanzstelle

Teilgebiet: Informatik im Bankwesen

Eine Instanz, die Dienstleistungen anbietet und eine Karte als Zahlungsmittel bzw. als Berechtigungsnachweis akzeptiert. Martiny

akzeptierbare Menge → *Menge, akzeptierbare*

Akzeptor, endlicher

Teilgebiet: Automatentheorie und formale Sprachen

finite state acceptor

Synonym: Automat, endlicher erkennender

Ein (endlicher) *Automat* heißt (endlicher) Akzeptor, wenn er Anfangs- und Endzustände besitzt, aber keine Ausgaben produziert. Die Menge aller

Akzeptor, Leistung des

endlichen Eingabefolgen, die den Automaten von einem Anfangs- in einen Endzustand überführen, heißt die Leistung des Akzeptors oder die vom Akzeptor akzeptierte Menge (oder Sprache)

$$A = (S, X, \delta, s_0, F)$$

mit dem Zustandsalphabet S , dem Eingabealphabet X , s_0 als Anfangszustand und $F \subseteq S$ als Menge der Endzustände sowie der Überföhrungsfunktion (\rightarrow *Übergangsfunktion*) $\delta: S \times X \rightarrow S$.

Die von A erkannte Wortmenge

$$T(A) = \{w \in X^* \mid \delta^*(w, s_0) \in F\}$$

ist eine reguläre Wortmenge.

δ^* ist die erweiterte Überföhrungsfunktion (*sequentielle Maschine*). Endliche Akzeptoren erkennen genau die regulären (\rightarrow *Menge, reguläre*) Wortmengen. Böhling

Akzeptor, Leistung des \rightarrow *Akzeptor, endlicher*

Akzeptor, m-adischer \rightarrow *Sprache, stochastische*

Akzeptor, stochastischer \rightarrow *Sprache, stochastische*

Alarm \rightarrow *Ereignis in einem Datenverarbeitungssystem*

Alarmierungszeit

Teilgebiet: Grundlegende Definitionen

Zeit zwischen dem Erkennen des Fehlers und der Alarmierung des für die Störrungsbehebung Verantwortlichen. Hammann-Kloss

Algebra, boolesche

Teilgebiet: Logik

Ein mathematisches System von Rechenregeln für *binäre Variablen*, benannt nach dem englischen Mathematiker George Boole (1815-1864). Burkhardt

Algebra, heterogene \rightarrow *Algebra, Σ -*

Algebra, homogene \rightarrow *Algebra, Σ -*

Algebra, Σ -

Teilgebiet: Theorie der Semantik

Eine Σ -Algebra $A = ((s^A)_{s \in S}, (f^A)_{f \in F})$ besteht aus einer endlichen Familie von Trägemengen s^A und einer endlichen Familie von Operationen f^A zwischen diesen Trägemengen. Das Paar $\Sigma = (S, F)$ heißt Signatur der Algebra A . Die Symbole $s \in S$ heißen *Sorten*, die Symbole $f \in F$ nennt man Operationssymbole, im Falle der *fixschreibweise* auch Verknüpfungssymbole. Jedem Operationssymbol $f \in F$ ist eine endliche Funktionalität $s_1 \times \dots \times s_n \rightarrow s_{n+1}$ mit Sorten $s_i \in S$ zugeordnet ($n \geq 1$). Diese Funktionalität legt den *Definitionsbereich* $s_1^A \times \dots \times s_n^A$ und den *Werte-*

bereich s_{n+1}^A der Operation f^A fest. Nullstellige Operationen heißen auch *Konstante*, zweistellige Operationen in Infixschreibweise auch Verknüpfungen. Wenn die Operationen f^A partiell definiert sind, nennt man A eine partielle Σ -Algebra. Im Falle einer totalen Abbildung f^A spricht man auch von einer Funktion anstatt von einer Operation. Hat eine Algebra A nur eine Sorte, so nennt man sie homogen; die Operationen f^A bilden dann Tupel (\rightarrow *n-Tupel*) von Elementen aus s^A wieder in ein Element aus s^A ab. Hat eine Algebra A mehrere Sorten, so heißt sie heterogen. Parsch

algebraischer Typ \rightarrow *Datentyp, algebraischer*

Algebra of Communicating Processes

Teilgebiet: Nebenläufige Prozesse

Abkürzung: ACP

Eine algebraische Prozeßtheorie (s. *Prozeßalgebra*) im Stile von CCS (\rightarrow *Calculus Of Communicating Systems*). Merkmale (\rightarrow *Attribut*) von ACP sind Abstraktions- und Strukturierungsoperationen sowie eine synchrone (\rightarrow *Synchronität*) Kommunikationsfunktion zwischen zwei Prozessen (s. *Interprozeßkommunikation, Operationen*). Bergmann; Leszak

ALGOL

Teilgebiet: Programmierung

ALGORithmic Language. Eine höhere Programmiersprache zur Beschreibung mathematischer und technisch-wissenschaftlicher Programme. Es sind zu unterscheiden: ALGOL 58 mit den Dialekten NELIAC, JOVIAL und MAD; ALGOL 60 bzw. REVISED ALGOL 60 als die eigentliche ALGOL-Sprache; und als spätere Neuentwicklung ALGOL 68. Burkhardt

ALGOL 58 \rightarrow *ALGOL*

ALGOL 60 \rightarrow *ALGOL*

ALGOL 68 \rightarrow *ALGOL*

algorithmische Programmiersprache \rightarrow *Programmiersprache, algorithmische*

algorithmische Programmierung \rightarrow *Programmierung, algorithmische*

Algorithmische Sprache \rightarrow *Programmiersprache, algorithmische*

Algorithmus

Teilgebiete: Algorithmen und Komplexitätstheorie; Berechenbarkeit, rekursive Funktionen *algorithm*

Synonym: Berechnungsverfahren, effektives

Allgemeines (eindeutiges) *Verfahren* zur Lösung einer Klasse gleichartiger Probleme (z.B. zur Berechnung einer Funktion für verschiedene Argumente), gegeben durch einen aus elementaren Anweisungen an einen (menschlichen oder maschinellen) Rechner bestehenden Text. Für menschliche Rechner kann dies ein Text der Umgangssprache sein, für eine Datenverarbeitungsanlage ist jeder Algorithmus als Programm in einer geeigneten *Programmiersprache* anzugeben. Der Rechner erwartet die Vorgabe eines speziellen *Problems* der Klasse (Eingabe, Argument), führt dann die Anweisungen des Algorithmus schrittweise durch, wobei die jeweils als nächste auszuführende Anweisung durch den Text des Algorithmus, die Eingabe und evtl. schon berechnete Zwischenergebnisse eindeutig festgelegt ist, und liefert nach endlichen vielen Schritten ein Ergebnis (*Ausgabe, Resultat*). Ist das Problem für gewisse spezielle Argumente nicht definiert, so bricht die Berechnung nicht ab.

Ergänzungen: Beispiele:

- a) Euklidischer Algorithmus zur Bestimmung des größten gemeinsamen Teilers.
- b) Gaußsches Verfahren zur Lösung linearer Gleichungssysteme.
- c) d'Hondtsches Verfahren zur Bestimmung der Sitzverteilung im Parlament aufgrund der Stimmenverteilung. Müller

Algorithmus, asymmetrischer → *Verschlüsselungsverfahren, asymmetrisches*

Algorithmus, genetischer

Teilgebiet: Algorithmen und Komplexitätstheorie
genetic algorithm

Algorithmen zur Lösung von angewandten Problemstellungen, z.B. von nicht gutartigen Optimierungsproblemen, die in den Bereich der sog. *Evolutionsstrategien* fallen, Konzeptionell basieren sie auf geeigneten Rekonstruktionsverfahren zur Erzeugung neuer „Populationen“ von Lösungen aus bereits vorhandenen „Populationen“ von Lösungen, wobei als besser eingestufte Lösungen einer früheren Stufe tendenziell einen höheren Einfluß auf die Gestaltung späterer Generationen haben als weniger erfolgreiche Lösungen. Der konkrete Einfluß basiert weniger oder gar nicht auf Einsicht in den Erfolg von Lösungen, sondern nur auf deren Bewertung (→ *Informationssysteme, Bewertung von*); er ist in der Regel *stochastisch* und schließt zusätzliche zufällige Veränderungen (ähnlich zu Mutationen im biologischen Bereich) mit ein. Radermacher

Algorithmus, kryptographischer

Teilgebiet: Informatik im Bankwesen

Siehe auch: Kerkhoffs' Prinzip

Berechnungsverfahren, das mittels eines Schlüssels *Kryptogramme* erzeugt. Martiny

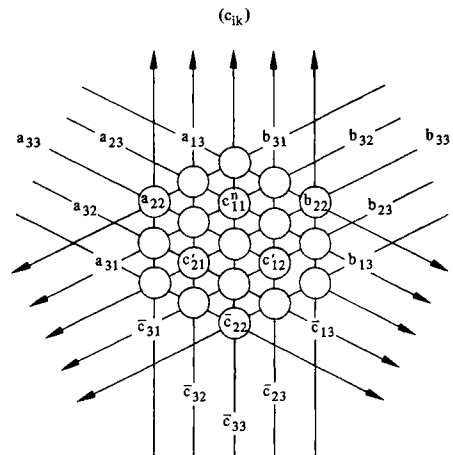
Algorithmus, symmetrischer → *Verschlüsselungsverfahren, symmetrisches*

Algorithmus, systolischer

Teilgebiet: Algorithmen und Komplexitätstheorie

algorithm, systolic

Ein synchron (→ *Synchronität*) paralleler Algorithmus für ein systolisches Feld. Ein typischer systolischer Algorithmus ist der zur Multiplikation von $n \times n$ -Matrizen (a_{ik}) und (b_{ik}) auf einem hexagonalen Feld von *Prozessoren*, bei dem jeder Prozessor an der Berechnung des Skalarproduktes c_{ik} einer *Zeile* und einer *Spalte* wie folgt mitwirkt: In einem *Takt* bestimmt der Prozessor das *Produkt* zweier Elemente (die aus zwei Richtungen in den Prozessor „gepumpt“ werden) und addiert das *Resultat* zu der aus der dritten Richtung kommenden Teilsumme des Skalarproduktes. Im nächsten Takt transportiert er die erhaltenen Matrixelemente und den Anteil des Skalarproduktes in der gleichen Richtung weiter zum jeweils nächsten Prozessor. Der Algorithmus arbeitet in Linearzeit.



$$c_{11}^n = a_{11} \cdot b_{11} + a_{12} \cdot b_{21}, c_{ik}^n = a_{i1} \cdot b_{1k}, \bar{c}_{ik} = 0$$

Systolische Algorithmen wurden außer für *Operationen* der linearen *Algebra* auch für die schnelle *Fourier-Transformation*, für die Signal-, Bild- und Sprachverarbeitung sowie für viele nichtnumerische Anwendungen (Such- und

Algorithmus, verteilter

Sortierverfahren (→ *Sortierung*), *Graphenalgorithmen*, *Codierungen* etc.) entwickelt. Brauer

Algorithmus, verteilter

Teilgebiete: Algorithmen und Komplexitätstheorie; Programmierung; Rechnernetze und verteilte Systeme

Ein verteilter Algorithmus ist ein Algorithmus, der ohne globale Steuerung auf einem verteilten System abläuft, wobei also jeder der verteilten *Prozessoren* Informationen, die für den Ablauf des Algorithmus nötig sind, nur aus einer beschränkten Umgebung erhält. Beispiele für verteilte Algorithmen sind: Verteilte Terminierung (d.h. Feststellung, ob im verteilten System alle Prozessoren ihre Aufgabe erledigt haben) oder die Wahl eines Anführers (Bestimmung eines ausgewählten Prozessors, der Steuerungsaufgaben für das gesamte System übernehmen soll).

Brauer

Algorithmusfehler → *Betriebsart eines Rechensystems*

Algorithmus zur Beseitigung verdeckter Linien

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

hidden line algorithm

Synonym: *Visibilitäts-Algorithmus*

Verfahren zur Entfernung derjenigen Liniensegmente eines Bildes, die normalerweise bei der zweidimensionalen Projektion eines dreidimensionalen Objekts verborgen sind. Es ist also ein Algorithmus, mit dem *Visibilitätskriterien* geprüft werden. Einige der wichtigen Algorithmen sind:

Appel-Algorithmus

Prioritätsverfahren

Überlagerungsverfahren

Warnock-Verfahren

Encarnação; Grieger

Allgemeine-Erlang-Verteilung

Teilgebiet: Messung, Modellierung und Bewertung von Rechensystemen

general Erlang distribution; phase type distribution; cox distribution

Synonyme: Cox-Verteilung; Gemischte-Erlang-Verteilung; Phasenverteilung, allgemeine

Familie von *Verteilungsfunktionen*, bestehend aus einer Summe gewichteter *Erlang-Verteilungen*. Cox zeigt, daß jede Verteilungsfunktion mit rationaler L-Transformierten als Allgemeine-Erlang-Verteilung *dargestellt* werden kann (als Sonderfälle sind die Exponential-, Erlang- und *Hyperexponentialverteilung* enthalten).

Diese Eigenschaft erlaubt es, alle in der Praxis

auf tretenden An kunfts- und *Bedienungsprozesse* mit beliebiger Genauigkeit nachzubilden. Die Allgemeine-Erlang-Verteilung kann (wie die Erlang- und Hyperexponentialverteilung) interpretiert werden als eine „Kombination von fiktiven *exponentiellen Phasen*“. Diese Modellvorstellung vereinfacht die mathematische Behandlung realer Prozesse wesentlich (Phasenmethode).

Herzog

allgemeiner Problemlöser → *Problemlöser, allgemeiner*

allgemeines graphisches System → *graphisches System, allgemeines*

allgemeine Systemtheorie → *Systemtheorie, allgemeine*

allgemeingültige Formel → *Formel, allgemeingültige*

Allgemeingültigkeit → *Formel, Gültigkeit einer*

ALLOCATE → *Speicherverwaltung, dynamische*

Allquantor → *Quantor*

Alltagswissen

Teilgebiet: Grundlegende Definitionen

Siehe auch: *Wissen*

Alltagswissen ist ein Wissen, das sich durch eine besonders hohe Kontextgebundenheit (bzw. „Situiertheit“) und/oder durch einen besonderen *Mangel* an Geltungsansprüchen auszeichnet. Alltagswissen ist schwer zu formalisieren und muß sich nicht in allen Fällen einem methodisch einlösbaren Anspruch an Wahrheit stellen: Es unterscheidet sich gelegentlich nur durch seine Gewissheitsansprüche von anspruchslosen Meinungen, die ihre *Quelle* in Traditionen oder ungeprüften Mitteilungen anderer Menschen besitzen. Um Alltagswissen, das beispielsweise auch in Polizei-, Kunden- und Personaldaten gespeichert wird, trotz seiner im Vergleich zum objektiven Wissen methodischen Defizite erfolgreich nutzen zu können, benötigt man *heuristische* Regeln, die die Bildung von Analogieschlüssen oder von rein qualitativ (also nicht statistisch (→ *Statistik*)) abgestützten Korrelationen zulassen. Jedermann bedient sich solcher heuristischer Mittel im Alltagsleben, jedoch ist es bisher noch nicht recht gelungen, ein wissenstechnisches System dahin zu bringen, die in ihm gespeicherten Bestände an Alltagswissen nach Regeln des „plausiblen Schließens“ zu organisieren und auszuwerten. Im Rahmen der Bemühungen um *künstliche Intelligenz* ist hinreichend deutlich geworden, daß bei der Ausprägung und Entwick-

lung von *Intelligenz* nicht nur dem lehr- und lernbaren wissenschaftlichen (Bildungs-)Wissen (\rightarrow Wissen, objektives), sondern auch dem vorwissenschaftlichen Alltagswissen eine sehr große Bedeutung zukommt. Literatur: H.L. Dreyfus, Was Computer noch immer nicht können, Dt.Z.f.Phil., 41 (1993), S.653-680; R. Kötter, Technische Rationalität und rationale *Heuristik* - Ein Problemaufriß, in: C. Burrichter / R. Inhetveen / R. Kötter (Hrsg.), Technische Rationalität und rationale Heuristik, Paderborn 1986, S.6-16; J.T. Schwartz, The Limits of Artificial Intelligence, in: St.C. Shapiro (Hrsg.): Encyclopedia of Artificial Intelligence, New York 1987, S.488-502; Luft; Richter, M.

Allzweckrechner

Teilgebiet: Rechnerarchitektur
general purpose computer

Ein Rechnersystem, das für beliebige Anwendungen konzipiert ist. Giloi

ALOHA-Protokoll

Teilgebiet: Rechnernetze und verteilte Systeme
ALOHA protocol

Dieses *Mehrfachzugriffsprotokoll* entstand bereits 1970 und wurde ursprünglich für Erdfunk- und Satellitennetze entwickelt. Die Grundidee dieses Verfahrens ist aber in allen *Mehrfachzugriffsnetzen* mit beliebigem Übertragungsmedium anwendbar.

Jede Station darf unabhängig vom Sendeverhalten anderer Stationen zu beliebigen Zeitpunkten ihre Daten *senden*. Bei *Kollision* zweier oder mehrerer zugleich stattfindender Sendevorgänge versucht jede betroffene Sendestation (\rightarrow *Sendebetrieb*) das erneute Senden der Daten zu einem späteren, zufällig bestimmten Zeitpunkt. Hierbei können wieder Kollisionen auftreten, die bei steigender *Belastung* das Protokoll instabil werden lassen. Halfar

ALPHA

Teilgebiet: Grundlagen von Informationssystemen

Die *Datenmanipulationssprache* ALPHA, die von CODD entwickelt wurde, ist eine *deskriptive Sprache*, die auf dem *Prädikatenkalkül* in Form des *Relationenkalküls* beruht. Die Sprache kann als *Teilsprache* (sublanguage) innerhalb einer *Gastgebersprache* benutzt werden. Die Sprache ALPHA wird hier stellvertretend für andere Sprachen, die auf dem Relationenkalkül basieren, genannt. Schneider

Alphabet

Teilgebiet: Grundlegende Definitionen

Ein linear geordneter Zeichenvorrat. Die über den Zeichenvorrat gelegte *Ordnungsrelation* (Anordnung der Zeichen) kann z.B. durch Vorbild, Absprache, Überlieferung oder Gewöhnung vorgegeben sein.

Alphabet und Zeichenvorrat unterscheiden sich durch diese Ordnungsrelation, die formal erklärt ist: In einem Zeichenvorrat A existiert genau dann eine totale *Ordnung*, wenn für alle $a, b \in A$ mit $a \leftarrow b$ entweder $a < b$ oder $b < a$ gilt; das Zeichen „ $<$ “ bedeutet die Relation „ist Vorgänger von“. Beispiele sind das aus *Buchstaben* bestehende Alphabet einer natürlichen Sprache, das *binäre* Alphabet oder das *sedezimale* Alphabet (nach DIN 44 300 T2).

Fleischhauer; Guse; Mätzel; Rouette

Alphageometrik

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

Bei diesem *Verfahren* wird der *Bildschirm* nicht in Schreibstellen unterteilt, sondern mit geometrischen Funktionen beschrieben. Die geometrischen Grundfunktionen wie *Linie*, *Kreis* oder *Polygon*zug können mit unterschiedlichen *Parametern* zur Herstellung von Grafiken und *Buchstaben* verwendet werden. Fuchs

Alphamosaik

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

Der *Bildschirm* wird in eine feste Anzahl von Zeilen mit einer festen Anzahl von Zeichen zur *Darstellung* von Texten aufgeteilt und Grafiken werden mosaikartig, d.h. aus einer geeigneten Anordnung von *Pixeln*, zusammengesetzt. Fuchs

alphanumerisch

Teilgebiet: Grundlegende Definitionen
alphanumeric

Eine Eigenschaft, die man hauptsächlich einem *Zeichenvorrat*¹ zuordnet und derzufolge dieser Zeichenvorrat mindestens aus den Dezimalziffern (\rightarrow *Dezimalsystem*) und *Buchstaben* besteht. Daneben wird sie aber auch als Eigenschaftsbezeichnung für Zeichenfolgen (\rightarrow *Zeichenkette*), Daten, Texte u.ä. benutzt.

Da diese *Definition* den Umfang des Zeichenvorrats offen läßt, ist es zweckmäßig, den Umfang des alphanumerischen Zeichenvorrats jeweils festzulegen. Beispielsweise können im Einzelfall *Sonderzeichen* dazu gehören.

Die Anwendung arithmetischer *Operationen* auf alphanumerische Daten ist in aller Regel nicht sinnvoll, dagegen aber die Anwendung von *Operationen* der *Verarbeitung* von Zeichenketten wie

alphanumerisches Zeichen

in der Text- oder Listenverarbeitung. Dazu gehören z.B. Vergleichs-, Verkettungs- oder Sortier-Operationen.

Entsprechend nennt man aus Ziffern und Buchstaben gebildete Zeichenanordnungen (Daten, Information, Texte usw.) alphanumerisch – im Unterschied zu rein *numerischen* und rein *alphabetischen* Darstellungen. Bei der üblicheren weiteren Begriffsauslegung enthalten alphanumerische Darstellungen auch Sonderzeichen (nach DIN 44 300 T4).

Fleischhauer; Gehring; Guse; Nilsson; Rouette

alphanumerisches Zeichen → *alphanumerisch*

ALPHARD → *Spezifikation*

Alternative → *Selektion*

Alternative, Auswahl → *Selektion*

ALW

Teilgebiet: Informatik im Bankwesen

Abkürzung von: Always

Gilt ALW als *Zugriffsbedingung* für ein bestimmtes *Kommando* (z.B. *lesender* oder *schreibender Zugriff*), unterliegt dieses keinerlei Restriktion. *Nachweis* von *Berechtigungen* ist nicht erforderlich. *Martiny*

Always → *ALW*

AM → *Amplitudenmodulation*

American National Standards Institut

→ *ANSI*

American Standard Code for Information Interchange → *ASCII*

Amplitudenmodulation

Teilgebiet: Datenübertragung

amplitude modulation

Abkürzung: AM

Codierungsverfahren, das den *Binärwerten* 0 und 1 unterschiedliche Amplituden einer Trägerfrequenz (carrier) als *Signalparameter* zuordnet.

$$u = (u_0 \pm u_m) \sin \omega t$$

Mehrstufen-AM wird kaum verwendet.

Häufigste Variante mit teilweise unterdrücktem Seitenband ist die *Restseitenbandmodulation* (vestigial sideband modulation). *Matejka*

Amplitudenwert, 90%- → *Taktimpuls*

analog

Teilgebiet: Analog- und Hybridtechnik

analogous

Die ursprüngliche Bedeutung des Wortes ist „entsprechend“, „ähnlich“, „im gleichen Verhältnis“. Demnach heißen zwei oder mehrere unter-

schiedliche Systeme analog, wenn ihre Strukturen bzw. deren Funktionen einander entsprechen. Dies bedeutet im allgemeinen, daß sie – zumindest im Rahmen der für die betrachteten Zusammenhänge relevanten Fragestellungen – der gleichen mathematischen Beschreibung genügen.

In der Datenverarbeitung und in der Kybernetik (→ *Regelungstheorie*) wird das Wort „analog“ häufig in der speziellen Bedeutung des Gegensatzpaares analog/digital bzw. analog/diskret verwendet. Die Bedeutung von „analog“ ist in diesem Zusammenhang auf „stetig veränderbar“ eingeengt. So versteht man z.B. unter einem analogen System ein System, dessen *Zustandsvektoren* in einem begrenzten oder unbegrenzten Bereich des *Zustandsraums* stetig veränderbar sind. Ein *analoges Signal* ist eine durch Zustandsänderungen des Informationsträgers (→ *Datenträger*) (Signalträger) dargestellte Zeitfunktion, deren *Wertebereich* einem Intervall des R_1 entspricht, d.h. beliebige Zwischenwerte enthält. *Rzehak*

Analogausgabe (-Einheit)

Teilgebiet: Prozeßdatenverarbeitung

analog output (unit)

Die Funktionseinheit eines *Prozeßrechensystems* zur Ausgabe analoger Signale. *Kopetz; Lauber*

Analog-Digital-Konverter → *Analog-Digital-Umsetzer*

Analog-Digital-Umsetzer

Teilgebiete: Analog- und Hybridtechnik; Prozeßdatenverarbeitung

analog digital converter

Synonyme: AD-Wandler; Analog-Digital-Konverter; Analog-Digital-Wandler

Eine Funktionseinheit, die ein *analoges* Eingangssignal in ein digitales Ausgangssignal umsetzt (DIN 19226). Dabei geht ein Anteil an Genauigkeit verloren, wenn der Wert in analoger Form zwischen den diskreten Wertmöglichkeiten der digitalen Form oder außerhalb dieser liegt.

Burkhardt; Lauber

Analog-Digital-Umsetzung

Teilgebiet: Analog- und Hybridtechnik

analog digital conversion

Synonym: Analog-Digital-Wandlung

Umsetzung eines *analogen* Eingangssignals in ein digitales Ausgangssignal. Diese Umsetzung ist zur digitalen Weiterverarbeitung der ursprünglichen analogen Signale (z.B. in *Prozeßrechnern*) erforderlich. Da im allgemeinen das Eingangssignal nicht streng konstant ist, hat das verwendete *Verfahren* einen gewissen Einfluß auf den digitalen Ausgangswert. Unter den ver-

schiedenen Verfahren zur Analog-Digital-Umsetzung unterscheidet man zwei Gruppen:

a) Augenblickswertverfahren; diese liefern den Meßwert (MW)

$$MW_{dig} = MW_{an}(t_i)$$

Der Zeitpunkt t_i heißt auch Abtastzeit.

b) Integrierende Verfahren; diese liefern den Meßwert

$$MW_{dig} = \frac{1}{T} \int_{t_0}^{t_0+T} MW_{an}(t) dt.$$

Ändert sich der Meßwert nur langsam und ist diesem eine periodische Funktion als Störung überlagert, so wird diese eliminiert, wenn für T die Periodendauer der Störung gewählt wird.

Rzehak

Analog-Digital-Wandler → Analog-Digital-Umsetzer

Analog-Digital-Wandlung → Analog-Digital-Umsetzung

analoge Datenverarbeitungsanlage → Analogrechner

Analogeingabe (-Einheit)

Teilgebiet: Prozeßdatenverarbeitung
analog input (unit)

Die Funktionseinheit eines Prozeßrechnensystems, mit der analoge Signale von außen zugeführt werden.

Kopetz; Lauber

analoge Rechanlage → Analogrechner

analoge Schaltung → Schaltung, analoge

analoges Signal → Signal, analoges

Analogierechenautomat → Analogrechner

Analogierechenmaschine → Analogrechner

analogisch

Teilgebiete: Kognitive Methoden – Künstliche Intelligenz; Wissensverarbeitung, Expertensysteme
analogical

Um eine Verwechslung mit der in der Nachrichtentechnik üblichen eingeschränkten Bedeutung des Begriffs analog zu vermeiden, wird in der KI in Anlehnung an den englischen Begriff analogical häufig die Bezeichnung analogisch verwendet, der auch bei Kant zu finden ist. Analogisch bezeichnet eine Ähnlichkeitsbeziehung zwischen zwei Strukturen. In der KI bezeichnet man eine Darstellung von Wissen als analogisch, wenn sie durch eine monotone Abbildung aus der dargestellten Struktur hervorgeht. Dabei ist es irrele-

vant, ob kontinuierliche oder diskrete Strukturen dargestellt werden. Analogische Darstellungen besitzen u.a. folgende Eigenschaften:

- 1) Kohärenz: Jedes Element einer dargestellten Situation erscheint nur einmal, wobei alle dargestellten Beziehungen unmittelbar zugänglich sind.
- 2) Stabilität: Kleine Abweichungen in der dargestellten Struktur bewirken keine großen Abweichungen in der dargestellten Struktur.
- 3) Simulationsfähigkeit: Analogische Modelle eignen sich aufgrund der Ähnlichkeit ihrer Struktur mit der dargestellten Situation zu deren Simulation.

Beispiele für analogische Darstellungen sind:

- eine zweidimensionale optische Projektion eines dreidimensionalen Gebildes
- eine Liste, deren Elemente entsprechend der Reihenfolge dessen geordnet sind, was sie darstellen.

Freksa

Analogrechner

Teilgebiet: Analog- und Hybridtechnik
analog computer

Synonyme: Analogierechenautomat; Analogierechenmaschine

Rechanlage, bei der zur Wertdarstellung der Rechengrößen ein analoges Signal verwendet wird, d.h. eine physikalische Größe, deren Wert der Rechengröße proportional ist. Als physikalische Größe wird heute nahezu ausschließlich die elektrische Spannung verwendet (Gleichspannungsanalogrechner). Auf diesem Rechner löst man ein mathematisches Problem dadurch, daß aus einer größeren Anzahl verschiedener elektronischer Rechelemente zur Realisierung der mathematischen Operationen ein physikalisches Modell des Problems aufgebaut wird, in welchem alle Rechengrößen des Problems als elektrische Spannung dargestellt sind. In diesem elektrischen Modell verhalten sich nun alle Rechenanspannungen in genau der gleichen Weise, wie dies für die Variablen (→ Variable!) des zu untersuchenden mathematischen Problems der Fall ist. Problemgrößen und Rechenanspannungen lassen sich exakt ineinander umrechnen. Das Programmieren des Analogrechners besteht im Herstellen der Verbindungen zwischen den Rechelementen (Rechenschaltung). Da die Integration über der Zeit als unabhängige Variable durch ein Rechelement (Integrierer) realisiert werden kann, eignet sich der Analogrechner insbesondere zur Lösung von Differentialgleichungen. Da hierzu ein elektrisches Modell verwendet wird, ist ein bevorzugtes Anwendungsgebiet die Simulation dynamischer Systeme. In einem Ana-

Analogspeicher

logrechner laufen alle Rechenoperationen (\rightarrow *Rechenprozeß*) parallel (\rightarrow *parallel*²), also gleichzeitig ab. Die zur Lösung des Problems benötigte *Rechenzeit* ist nicht abhängig vom Umfang des Problems, sondern wird nur von der Rechengeschwindigkeit der einzelnen Rechenelemente bestimmt. Dagegen ist der Umfang der Reenschaltung der Größe des Problems direkt proportional.

Rzehak

Analogspeicher

Teilgebiet: Analog- und Hybridtechnik

analog memory

Ein *Speicherelement* zur Erhaltung von Daten in *analoger Darstellung* (meist als Spannungen), z.B. aus Kondensatoren aufgebaut.

Burkhardt

Analogssteuerung \rightarrow *Steuerung*

Analogschnik

Teilgebiet: Analog- und Hybridtechnik

analog instrumentation

Bereich der Technik, der sich mit Entwurf und Aufbau solcher Teile von Steuerungen und Regelungen oder Datenverarbeitungsgeräten beschäftigt, in denen die Informationsdarstellung und -verarbeitung mit Hilfe *analoger* Signale geschieht.

Rzehak

Analyse

Teilgebiet: Informatik in der Betriebswirtschaft

Analyse ist die Zerlegung eines als System aufgefaßten Ganzen in seine einzelnen Teile. Als *Systemanalyse* dient sie allgemein der organisatorischen Gestaltung von *Betrieben*. Die Analyse kann in folgende Stufen gegliedert werden:

1. Analyse als Zielsetzung
2. Analyse der Elemente
3. Analyse der Beziehungen
4. Analyse des Systemverhaltens (\rightarrow *System*¹).

In der Datenverarbeitung stellt die Systemanalyse die Voraussetzung für die Systemgestaltung dar, indem sie eine Aufgabenstellung hinsichtlich des Istzustands (\rightarrow *Istanalyse*), der einzusetzenden Methoden und *Verfahren*, der benötigten Informationen sowie der Form und des Inhalts der Ausgabedaten untersucht und eine Lösung erarbeitet, die der Programmierung zugeführt werden kann.

Als weitere Objekte der Analyse als Voraussetzung einer organisatorischen Gestaltung können die Arbeitsanalyse und die Aufgabenanalyse genannt werden.

BIFOA

Analyse, lexikalische

Teilgebiet: Übersetzerbau

Die lexikalische Analyse ermittelt in einer einen Programmtext repräsentierenden *Zeichenkette* die Grundsymbole der zugehörigen *Programmiersprache*. Innerhalb eines *Compilers* bereitet der *lexikalische* Analytiker (\rightarrow *Analyse*) den Quelltext für die syntaktische (\rightarrow *Syntax von Programmiersprachen*) Analyse auf. Seine Aufgaben sind die Entfernung bedeutungsloser Zeichen, die *Klassifizierung* der *Trennsymbole*, der Aufbau des *Bezeichnerlexikons* und die Berechnung von *Zahlwerten*.

Lehmann

Analyse, linguistische \rightarrow *Sprachanalyse, automatische*

Analyse, morphologische

Teilgebiet: Information Retrieval

analysis, morphological

Zerlegung eines Wortes in seine Morpheme und Zuordnung dieser Morpheme zu linguistischen Kategorien, also zu Klassen von Elementen mit gleichen Merkmalen (\rightarrow *Attribut*). Hierzu werden beispielsweise stringorientierte *Verfahren* (stringbezogene Analysealgorithmen) oder lexikonorientierte Verfahren auf der Basis möglichst vollständiger, spezieller Fachlexika oder morphemorientierter Verfahren (lexikonorientierte Morphemlisten in Kombination mit morphemorientierten Analysemethoden) eingesetzt.

Freiburg

Analyse, objektorientierte¹

Teilgebiet: Programmierungstechnik

Das *objektorientierte* Paradigma (\rightarrow *Programmierparadigma*) verwendet zur Beschreibung und Entwicklung von *Informationssystemen* bestimmte Konstrukte wie Objekt, Klasse, *Attribut* und Methode sowie die Möglichkeit, über Vererbungs- und Verwendungsbeziehungen den Zusammenhang zwischen Klassen zu spezifizieren. Der Einsatz objektorientierter *Programmiersprachen* (z.B. C++, *Smalltalk*, *Eiffel*), die sich dieser Konzepte bedienen, hat sich in vielen Anwendungsbereichen als sehr vorteilhaft erwiesen.

Dies hat dazu geführt, daß mit einem gewissen Zeitverzug die Frage diskutiert wurde, wie Konzepte des objektorientierten Paradigmas auch in den frühen *Phasen* der Entwicklung (*Analyse* und *Design*) Berücksichtigung finden können, da man sich durch die Verwendung gleicher Beschreibungskonstrukte auf allen Ebenen eine hohe Durchgängigkeit der Entwicklung und einfachere Abbildung der Realität versprach.

In den letzten Jahren hat eine Inflation objektorientierter Analyse- und Designmethoden stattgefunden, wobei jeweils zu unterscheiden ist, ob es dabei um reine *Notationen* oder auch Vorgehens-

modelle handelt. Auch wenn eine *Standardisierung* noch nicht zu erkennen ist, haben in der Praxis doch einige Methoden besonders häufige Anwendung gefunden, wie z.B. Object Modeling Technique (Rumbaugh et al.), Object-Oriented Analysis and Design (Booch) oder Object-Oriented Analysis (Coad, Yourdon). *Heß*

Analyse, objektorientierte²

Teilgebiet: Programmierungstechnik

Methode der *Definition von Anforderungen* an ein *Informatiksystem*, wobei die funktionalen Anforderungen und die relevanten Daten in Form von Objekten und ihren Beziehungen zueinander *dargestellt* (modelliert (→ *Modellierung*)) werden. Dabei wird versucht, solche Objekte zu wählen, die mit realen Objekten, wie sie in der Anwendungswelt vorkommen, möglichst gut korrespondieren. *Endres*

Analyse, semantische

Teilgebiet: Linguistische Datenverarbeitung
semantic analysis

Die semantische Analyse (→ *Erschließung, inhaltliche*) versucht, den Inhalt einer sprachlichen Äußerung zu bestimmen. Für die automatische Durchführung ist daher eine Inhaltsdarstellungsmethode oder Methode zur *Darstellung von Wissen* (s. Repräsentationstheorie) erforderlich. Die Forschungsarbeiten auf diesem Gebiet sind trotz wichtiger Zwischenergebnisse noch lange nicht abgeschlossen. Die bedeutendsten Beiträge kommen aus der *Modelltheorie*, prozeduralen *Semantik* und Psycholinguistik. *Frenkel*

Analyse, semantische, bei Programmiersprachen

Teilgebiet: Übersetzerbau
semantic analysis of programming languages

Im *Compilerbau* versteht man unter semantischer Analyse (→ *Erschließung, inhaltliche*) eines Programmtextes die Bestimmung der bedeutungsrelevanten *Attribute* der durch eine *kontextfreie Grammatik* definierten syntaktischen (→ *Syntax von Programmiersprachen*) Spracheinheiten. Den vom syntaktischen Analysierer (→ *Parser*) erhaltenen *Ableitungsbaum* ergänzt der semantische Analysierer um die für die *Codeerzeugung* (→ *Codegenerator*) wichtige Information. Oft nimmt die *Ausgabe* der semantischen Analyse die Form einer linearen Kette von *Operationen* für eine *abstrakte Maschine* an, z.B. ADDIERE ALPHA (ganzzahlige *Konstante*) zu BETA (eindimensionales Feld von Gleitpunktzahlen (→ *Gleitkommaechnung*)), *Indexbereich*: 15 – 24). Typische Aufgaben der semantischen Analyse sind die *Prüfung*, ob *Operanden* unter-

einander und mit der Operation *typ* verträglich sind, und die Bestimmung von *Gültigkeitsbereichen* für *Bezeichner*. *Lehmann*

Analyse, strukturierte

Teilgebiet: Programmierungstechnik
structured analysis
Abkürzung: SA

Eine *Software-Entwicklungsmethode*, die als Modellvorstellung einen Datenflußgraphen benutzt, zusammen mit Minispecs (Beschreibung der *Knoten im Graphen*) und der Beschreibung von *Datenflüssen* und *Dateien* durch ein Data Dictionary (→ *Datenlexikon*). *Schneider*

Analyse, syntaktische, bei formalen Sprachen

Teilgebiet: Übersetzerbau
syntactical analysis

Im *Compilerbau* erfaßt man mit dem Begriff *Syntax* einer *Programmiersprache* nur die *kontextfreien* syntaktischen (→ *Syntax von Programmiersprachen*) Aspekte einer Sprache; diese beschreibt man durch eine Regelgrammatik (→ *Phrasenstrukturgrammatik*) in einer der Varianten der *Backus-Naur-Form*. Die syntaktische *Analyse* gliedert einen Programmtext gemäß den definierenden *grammatischen* Kategorien; sie erstellt den (bei Mehrdeutigkeit der Grammatik: einen) zugehörigen *Ableitungsbaum*. *Götler; Lehmann*

Analysebaum → Startsymbol

Analyse des Anwendungsgebiets → *Anwendungsanalyse*

Analysestrategie

Teilgebiet: Linguistische Datenverarbeitung

Aufgabe eines *Parsers* ist es, einer vorliegenden *Zeichenkette* – sei es einer *Programmiersprache*, sei es einer natürlichen Sprache – relativ zu einer *Grammatik* eine *Strukturbeschreibung* zuzuordnen (*Syntaxanalyse*). Diese Aufgabe läßt sich in Teilaufgaben aufspalten, zu deren Lösung es jeweils Alternativen gibt. Eine Analysestrategie ergibt sich aus einer bestimmten Kombination von Lösungen der folgenden Teilaufgaben.

(i) *Lexikonzugriff*: Es ist ein Wortformenlexikon möglich, bei dem die Wörter in der Eingabe unmittelbar mit den Einträgen im *Lexikon* verglichen werden, oder ein Stammformen- oder Grundformenlexikon, bei dem die eingegebenen Wörter einer morphologischen *Analyse* unterzogen werden müssen.

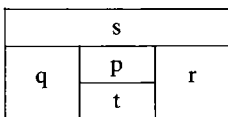
(ii) *Abarbeitung der Eingabe*: Die Eingabe wird von links nach rechts oder von rechts nach links durchlaufen. Es gibt einen oder mehrere Durch-

gänge durch die Eingabe oder die Abarbeitung der Eingabe erfolgt überhaupt nicht fortlaufend, sondern z.B. von bestimmten Stellen aus nach links und rechts (Insel-Parsing).

(iii) Form und Abarbeitung der Grammatik: Prinzipielle Möglichkeiten sind die Konstruktion einer Strukturbeschreibung durch Anwendung von *Produktionen (Ersetzungsregeln)*, das Erkennen der Struktur der Eingabe durch Vergleich mit *Mustern* oder *Übergangsnetzwerken*, der Aufbau einer Struktur durch Einsetzen von Teilstrukturen in *Leerstellen* (slot-filler-approach), die Ausführung von Prozeduren (→ *Unterprogramm*) bei in den *Parser* integrierten Grammatiken.

(iv) Erzeugung eines *Analysebaumes*: In der Regel hat die gesuchte Strukturbeschreibung die Form eines *Baumes*. Man kann den Baum von der *Wurzel* her (top-down (→ *Top-Down-Verfahren*)) oder von den *Blättern* her (bottom-up (→ *Bottom-up-Strategie*) (→ *Bottom-up-Verfahren*)) rekonstruieren (die Bezeichnungsweise rührt von der zeichnerischen Darstellung her, bei der die Wurzel des Baumes das *Startsymbol* der Grammatik und die Blätter des Baumes die Wörter der Eingabe sind, wobei das Startsymbol oben und die Wörter unten gezeichnet werden). Im ersten Fall werden die Produktionen der Grammatik dazu benutzt, vom Startsymbol ausgehend, die Kategorien durch ihre unmittelbaren Konstituenten zu ersetzen (zu expandieren), im zweiten Fall dazu, Kategorien zusammenzufassen (zu reduzieren), bis das Startsymbol erreicht ist.

(v) *Reihenfolge* der Produktionen: Die nächste Produktion wird immer auf die am weitesten links stehende (oder alternativ auf die am weitesten rechts stehende), noch nicht bearbeitete Kategorie angewendet. Dies ist das Prinzip „Tiefe zuerst“ (depth-first). Sei p der zu einer einzelnen Produktion gehörende Teilbaum; seien s, q, r und t zu beliebigen Teilableitungen gehörende Teilbäume:



In dem Augenblick, wo p aufgefunden wird, sind bei einem Top-Down-Verfahren die Teilbäume s und q vollständig rekonstruiert, bei einem Bottom-up-Verfahren dagegen q und t. Beim Prinzip „Breite-zuerst“ (breadth-first) erfolgt die weitere Abarbeitung von Kategorien in der Reihenfolge ihres Entstehens, d.h. die früher erzeugten zuerst, die später erzeugten später. Dies führt dazu, daß der Analysebaum immer auf ganzer Breite ausge-

füllt wird.

(vi) *Auswahl* der Produktionen: Die Analyseverfahren unterscheiden sich weiter darin, wie sie unter Berücksichtigung des Kontextes die Produktion p bestimmen. (→ *Parser*; Grammatik, *prädikative*).

(vii) *Verwaltung* der Ergebnisse: Es ist immer nur das aktuelle Gesamtergebnis gespeichert, nämlich der im Entstehen begriffene Analysebaum. Es kann passieren, daß Teile der aufgebauten Struktur revidiert werden müssen und daß dann beim Neuaufbau schon einmal geleistete Arbeit wiederholt werden muß. Alternativ werden alle Zwischenergebnisse bzw. Teilbäume in einer zentralen *Tabelle* gespeichert (well-formed substring table). Ein in einem Stadium erstelltes Zwischenergebnis kann in anderem Zusammenhang wieder verwendet werden (*Chart-Parser*).

(viii) *Verfahren* bei Alternativen: Ein Parser für natürliche Sprachen kann im Prinzip nicht vollständig *deterministisch* sein, da eine Äußerung z.B. gewollt mehrdeutig sein kann. Nicht-deterministische Zustände können jedoch deterministisch abgearbeitet werden. Es gibt dazu zwei Möglichkeiten: Eine *Alternative* wird so weit wie möglich verfolgt. Anschließend wird der Parser in den alten Zustand zurückversetzt (→ *Rücksetzen*) und die nächste Alternative wird verfolgt. Dies wird fortgesetzt, bis keine Alternative mehr unbearbeitet ist. Oder alle Alternativen werden gleichzeitig abgearbeitet (*Parallelverarbeitung*). Das heißt, daß die *Kontrolle* verzweigt und der *Automat* Mengen von Zuständen hat, bzw. daß eine Menge von Automaten an demselben *Problem* arbeitet. Hellwig

Analysetiefe

Teilgebiet: Linguistische Datenverarbeitung

An das Analyseergebnis eines Parsers können mehr oder weniger hohe *Anforderungen* gestellt werden.

(i) Im einen Extremfall verlangen wir nur, daß der *Parser* einen Text überfliegt (engl. skimming) und bestimmte linear abgrenzbare Einheiten entdeckt, wobei wir u.U. zulassen, daß nicht identifizierbare Teilstücke ignoriert werden (partielles Parsing (→ *Parsing online*)). Als *Muster* für die gesuchten Einheiten dienen reguläre Ausdrücke. Da eine solche lineare Durchmusterung der Eingabe in *Compilern* für *Programmiersprachen* dazu verwendet wird, die Einheiten für die eigentliche syntaktische *Analyse* bereitzustellen, wird sie auch „lexikalische Analyse“ genannt. Für Zwecke des *information retrieval* ist ein solches Überfliegen von Texten u.U. schon ausreichend.

(ii) Im anderen Extremfall wird eine vollständige

hierarchische Strukturierung der Eingabe relativ zu einer gegebenen *Grammatik* u.U. mit Berücksichtigung semantischer Phänomene verlangt.

Hellwig

Analyse- und Prognosesystem → *Prognosesystem*

Analysewörterbuch → *Maschinenwörterbuch*

Analysieren

Teilgebiet: Programmierungstechnik

Einen Sachverhalt analysieren heißt, durch Untersuchen genauere Kenntnisse über ihn zu gewinnen, z.B. durch detaillierte Betrachtung, Auflösung, Zerlegung in überschaubare Teileinheiten, Herausstellen von Einzelheiten. Das Ergebnis ist die Analyse.

Anmerkung: Analysen können in verschiedenen Aufgabenkomplexen der Herstellung und Anwendung von Software vorgenommen werden, z.B. kann bei der Aufgabendefinition eine Analyse der Anforderungen bzgl. der technischen Machbarkeit (Laufzeitverhalten, Speicherbedarf u. dgl.), bei der Realisierung eine Analyse von Fehlern, bei der Anwendungsbereitstellung eine Analyse der Einsatzbedingungen notwendig sein.

Fritsche

Analyst Workbench

Teilgebiet: Programmierungstechnik

Software-Werkzeuge für die *Analyse-* und *Spezifikationsphase* im *Software-Lebenszyklus*, insbesondere mit grafischer Benutzeroberfläche, *Data-Dictionaryfunktion* und der *Prototypingfähigkeit*.

Schneider

analytische Statistik → *Stochastik*

Anamnese, computerunterstützte

Teilgebiet: Informatik in der Medizin

Einsatz von EDV-Methoden zur Aufnahme der (med.) „Vorgeschichte des Kranken“ und seiner Beschwerden (= Anamnese).

Off-line: z.B. Verwendung von vorgedruckten *Lochkarten*, die vom Patienten in „JA“, „NEIN“, „ICH WEISS NICHT“-Fächer zu legen sind oder von Markierungsbelegen.

On-line: *Dialog* über *Datenstation* oder *PC*, zu bedienen vom Arzt oder auch vom Patienten.

Koeppel

AND → *Verknüpfungsglied*

Aneinanderreihung → *Reihung*

Anfangsadresse → *Basisadresse*

Anfangskonfiguration → *Kellerautomat*

Anfangsprädikat → *Prädikatumformer*

Anfangszeichen → *Startsymbol*

Anfangszusicherung → *Zusicherung*

Anforderung

Teilgebiet: Grundlegende Definitionen

Siehe auch: *Qualitätsforderung*; *Forderungen der Gesellschaft*

Festlegung, die zu erfüllende Kriterien (nach *DIN EN 45 020*, Ausgabe, April 1994) gibt.

Möller

Anforderungen

Teilgebiet: Programmierungstechnik
requirements

Anforderungen an ein System sind Aussagen über zu erbringende *Leistungen*. Sie lassen sich qualitativ unterscheiden in funktionale und nichtfunktionale Anforderungen.

Funktionale Anforderungen („Was soll das System aufgrund der Aufgabenstellungen können?“) umfassen

- Eingaben und deren Einschränkungen,
- Funktionen, die das System ausführen können soll,
- Ausgaben und sonstige Reaktionen des Systems.

Nichtfunktionale Anforderungen lassen sich weiter unterscheiden in

- a) *Qualitätsattribute* der gewünschten Funktionen („Wie soll das System die gestellten Aufgaben erfüllen?“)
 - Ausführungsverhalten (Zeit, Speicher)
 - Wartbarkeit (→ *Wartung*)
 - *Zuverlässigkeit* (Ausfallsicherheit / *Robustheit*, *Fehlererkennung* und -behandlung)
 - Sonstige Qualitätskriterien (*Portabilität*, *Flexibilität*, *Kompatibilität* mit vorhandenen Systemen)
 - Menschliche Faktoren (*Benutzerfreundlichkeit*, Qualifikation des Bedienpersonals (→ *Operateur*))
- b) Anforderungen an die *Realisierung* des Systems
 - Realisierung in *Software* und/oder *Hardware*
 - Geräte (→ *Peripheriegerät*)
 - Schnittstellen
 - Verwendung vorgegebener Hilfsmittel (*Programmiersprache*, *Betriebssysteme*, Rechner, sonstige Bausteine)
 - *Dokumentation* des Systems
- c) Anforderungen an *Prüfung*, Einführung und Betreuung
 - Testvorbereitung (→ *Testen*) und *Test*

Anforderungen an Schulprogrammiersprachen

- Abnahme
 - Freigabe / Endprüfung
 - Konfigurationsmanagement / Nachvollziehbarkeit
 - Kundendienst (Wartung / Änderung, Garantie, Archivierung, Ersatzteile, Schulung)
- d) *Anforderung an die Durchführung der Systemerstellung (→ System¹)*
- Globale Vorgehensweise
 - Zu verwendende Hilfsmittel (Methoden, Beschreibungsmittel, Werkzeuge)
 - Zur Verfügung stehende Ressourcen (Maschinenzeit (→ CPU-Zeit) / Kapazität / Konfiguration, verfügbare Mannschaft, Termine und sonstige zeitliche Beschränkungen, Kosten)
 - Zu berücksichtigende Vorschriften, Richtlinien, Normen
 - Dokumentation der Systemerstellung.

Partsch

Anforderungen an Schulprogrammiersprachen → *Schulprogrammiersprachen, Anforderungen an*

Anforderungen, funktionale → *Anforderungen*

Anforderungen, nicht-funktionale → *Anforderungen*

Anforderungsanalyse → *Baukastenprinzip*

Anforderungsbetrieb

Teilgebiet: Rechnerarchitektur

Ein *Betrieb eines Rechensystems*, bei dem eine Zentraleinheit von einer *Benutzerstation* zur Übernahme angebotener Daten veranlaßt wird.

Schneider

Anforderungsdefinition

Teilgebiet: Systemanalyse
requirements definition

Synonym: Bedarfsbeschreibung
Siehe auch: Aufgaben-Definition

Die Anforderungsdefinition ist jene Phase im *Phasenkonzept*, welche sich auf das „WAS“ eines Systems konzentriert. Daher wird im Gegensatz zur *Spezifikation*, die ihr zeitlich folgt, das „WIE“ einer konkreten Lösung möglichst ausgeklammert.

Praktische *Werkzeuge* sind aus den *Problembeschreibungssprachen* herausgewachsen und führen meist keine sehr scharfe Trennung von *Anforderung* und *Spezifikation* durch. So wird auch *PSL/PSA* als Mittel der *Anforderungsanalyse* eingesetzt. Manche Werkzeuge wie *SADT* (Ross, Schoman) ermöglichen es, die individuel-

len *Anforderungen* vieler Benutzer zu sammeln und zu integrieren.

Traummüller

Anfrage → *Suchfrage¹*

Angewandte Informatik

Teilgebiet: Grundlegende Definitionen
Applied computer science

Angewandte Informatik ist die etwa gleichzeitig zur sog. Kerninformatik entstandene Wissenschaft über die Anwendung der maschinellen *Verarbeitung* und Übermittlung von Informationen (siehe Informatik).

Erläuterung:

„Anwendung“ meint die Beziehung der Informatik (genauer: der Informatikpraxis) zu den gesellschaftlichen Teilbereichen (Staat, Wirtschaft, deren Teile) wie zur Gesellschaft als Ganzer. Angewandte Informatik wird notwendig, da weder Informatiker über das fachliche, noch Betriebswirte, Mediziner oder Verwaltungsfachleute über das technische Know-how zur sachgerechten Einführung von Informationstechnik-Systemen verfügen: In diesem Niemandland siedelt Angewandte Informatik. Ziel oder Aufgabe ist die sozial angemessene („sozialadäquate“) Einführung *computergestützter* Systeme. Sie schlägt damit die notwendige *Brücke* zwischen der überwiegend formalen Methodik der (Kern-) Informatik und den meist eher inhaltlichen und funktions-spezifischen *Anforderungen* der Anwendungsbe-reiche, ist also eine „Brückenwissenschaft“.

Maßstab (→ *Skalierung¹*) der Sozialadäquanz:

Sozialadäquat oder gleichbedeutend „sozialökologisch“ sei diejenige Einführung, die gemäß den Maßstäben und Standards der jeweils betroffenen gesellschaftlichen Teilsysteme (→ *System¹*) geschieht, insbesondere was den Vorgaben der *Verfassung* genügt (z.B. „informationelles Selbstbestimmungsrecht“); wobei die Gesellschaft ihrer-seits durch die Europäische Gemeinschaft, den Staat und durch die Beteiligten, aber auch durch spezielle Expertengremien der Technik Normen unterschiedlichster Art vorgibt, wie Kommunikationsprotokolle, Bildschirmnormen, *Datenschutzvorschriften*, Betriebsvereinbarungen oder Computerstrafrecht (→ *ADV-Recht, Computerkriminalität, -recht*).

Dementsprechend umfaßt sie i.w.S. die Theorie dieser Anwendungsgebiete (siehe: Produktions-, Medizinische, Rechts-, Verwaltungs-, Wirtschafts- usw. -informatik), d.h. alle anwendungsbereichsbezogenen Aspekte der Informatik, ist also eine Sammelbezeichnung für die Menge der Anwendungsinformatiken oder Angewandten Informatiken, auch „Fachinformatiken“ (inkorrekt „Anwendungen“) genannt.

Nun ergibt eine Summe noch kein wissenschaftliches Fach. Im engeren und eigentlichen Sinn umfaßt Angewandte Informatik die diesen Fachinformatiken gemeinsamen Teile. „Die“ Angewandte Informatik besteht aus diesem gemeinsamen Bestand, hält sie zu einem einheitlichen Ganzen zusammen, verbindet sie zur Kerninformatik und trägt deshalb eine eigene Bezeichnung. Die Angewandte Informatik bildet gleichsam den vor die *Klammer* gezogenen „Allgemeinen Teil“ der speziellen Informatiken, und ist als solcher von diesen zu unterscheiden:

Angewandte Informatik			
Angewandte (Fach-) Informatiken:			
Rechtsinformatik	Wirtschaftsinformatik	Medizinische Informatik	Sonstige Anwendungsinformatiken

Kontext:

Verhältnis zu „Informatik und Gesellschaft“ (IuG): Auf den ersten Blick fallen sie zusammen; „Angewandte Informatik“ als Wissenschaft über die Beziehung der Informatik zu gesellschaftlichen Bereichen und „Informatik und Gesellschaft“ scheinen synonym. Doch geht IuG aus praktischen Gründen weiter. Denn sie umfaßt auch die umgekehrte Beziehung der Anwendungsbereiche hin zur Informatik; etwa arbeitswissenschaftliche oder Produkthaftungs-Fragen bei Einführung von Informationstechnologien. – Selbst das ist ungenau: Selbstverständlich behandelt *Wirtschaftsinformatik* auch organisatorische, *Verwaltungsinformatik* auch verwaltungswissenschaftliche Fragen, aber eben nur soweit sie für Entwurf und *Implementierung von Informationssystemen* Besonderheiten aufweisen. – Angewandte Informatik ist demnach Querschnittswissenschaft: Sie bedient sich aller Disziplinen, Methoden und *Verfahren*, deren die Einführung und sozialadäquate Ausgestaltung / Einbettung von Informationstechnologie bedarf. Sie arbeitet deshalb transdisziplinär (über Fachgrenzen hinaus) und polymethodisch (mit ≥ 2 Methoden). Sie unterscheidet sich dadurch von der Kerninformatik und ähnelt in dieser Hinsicht besonders der Architekturwissenschaft, die sich aller Verfahren bedient, die dem Hausbau – hier dem Informationssystem-Bau – dienen. Ihr Leitbild ist darum auch nicht der bloße *Programmierer*, sondern der Informationssystem-Architekt.

Der Aufbau entspricht ihrer Aufgabe.

1. Ein Grundlagenteil hat die Aufgabe, wissenschaftstheoretisch die Identität und Eigenart

dieser projektorientierten, darum zugleich ingenieur- wie natur- und sozialwissenschaftlichen interdisziplinären Disziplin zu begründen, ihre theoretischen und sprachlichen Grundlagen zu legen, schließlich die Beziehungen und Grenzen zur Theoretischen und Praktischen Informatik, den Fachinformatiken sowie den Nachbar- und anderen Hilfswissenschaften zu bestimmen.

2. Grundbegriff ist gemäß der Aufgabe nicht *Berechenbarkeit* oder Algorithmus, sondern – im Anschluß an einige Naturwissenschaften – (System, Prozeß und) „Modell“; davon abgeleitet Information, Struktur und Organisation sowie deren Zusammensetzungen: Die *Funktionalität* von „Berechenbarkeit“ usw. kann die bei der Einbettung von realen Informationstechnik-gestützten Systemen auftretenden theoretischen und praktischen, insbesondere sozialen Probleme nicht hinreichend abbilden, da sie sich meist der Berechenbarkeit entziehen, und muß deshalb systemtheoretisch erweitert werden. – „Information“ wird in diesem Zusammenhang verstanden als Modell über ein Original für Zwecke von Zwecksetzern („Systemherrn“: Organisationen und Menschen, nicht nur Benutzer), also als vierstelliges „Modellwovon-wofür-für wen“, in seinen (im Anschluß an G. KLAUS) vier semiotischen Dimensionen: der syntaktischen (der formalen Daten, allgemeiner der Rechnerzustände; → Syntax von Programmiersprachen); der *semantischen* (der sprachlichen Bedeutungen); der pragmatischen (der Zwecke der Informationssystemerzeuger und Nutzer); schließlich der *sigmatischen* (des Realitätsbezugs und -gehalts der Information). – Dementsprechend ist das Informationssystem i.w.S. ein von einer zwecksetzenden Instanz definiertes System über zweckdienliche Information, das i.e.S. mit Computerunterstützung (also einschließlich eventueller telekommunikativer Anteile) arbeitet. Dieser „angewandte“ *Begriff* des Informationssystems, wie er zuerst von der Wirtschaftsinformatik entwickelt und dann von den anderen Fachinformatiken übernommen wurde, bezieht demnach neben der *Software* weitere Systemkomponenten mit ein: „Mensch“, „Information“ (in dem semantischen usw. Dimensionen), „Organisation“, „Systemzweck“ und „Außenbezüge“ (etwa Online-Verbindungen) mit ein, um deren Verhältnis zur Technik und Systemumgebung adäquat bestimmen zu können.

3. Die Methodik verbindet formale, inhaltliche und andere Elemente zu einem system- und risikoanalytischen Methoden- und Verfahrens-bündel, wie es ursprünglich zur Durchführung komplexer militärischer Projekte entwickelt, später allgemein auf Technikeinführungen ausgeweitet, schließlich im Rahmen der Verwaltungs- und besonders der Wirtschaftsinformatik im Hinblick auf Informationstechnik(einführung) spezialisiert wurde. In diesem weitgespannten Rahmen wird selbstverständlich die überwiegend formale („syntaktische“) Vorgehensweise der Kerninformatik integrierend berücksichtigt, ebenso wie spezielle Methoden der Betriebswirtschaft, der Rechtswissenschaft u.a. Interessant sind zum einen die Einbeziehung sozialwissenschaftlicher Erkenntnisse über unterschiedliche soziale Folgen unterschiedlich gestalteter Informationssysteme (woraus sich eine Informatikspezifische Informationstechnik-Folgenabschätzung ergibt!), ferner bestimmte Erweiterungen kerninformatischer Verfahren, etwa von der Software- zur Kontextspezifikation, oder von der *Hardware*- über die Software- zur Sozialergonomie.
4. Vor den Methoden stehen allerdings inhaltliche Ziele, anhand derer Angewandte Informatik nicht nur die zur Zielerreichung passenden Methoden und Verfahren auswählt, sondern sich vor allem der vorgeordneten Frage der sozialadäquaten *Auswahl* aus den n Gestaltungsmöglichkeiten des zu entwickelnden Systems annimmt. Was aber „paßt“? Entsprechend der übergeordneten und z.T. durch die Verfassung festgelegten Aufgabe der Sozialadäquanz technikgestützter Systeme sind maßgebende Kriterien des Systementwurfs nicht mehr allein das abstrakte *Pflichtenheft* mit den sich darin niederschlagenden ökonomischen bzw. politischen Zwecken (des jeweiligen *Betriebs* oder der auftraggebenden Behörde), sondern zusätzlich technische Sicherheits-, arbeitswissenschaftliche Ergonomie-, rechtliche Mitbestimmungs- und (überwiegend) organisatorische Datenschutz-Anforderungen wie schließlich das soziale Kriterium möglichstster Vermeidung sozialer Negativfolgen (z.B. Arbeitslosigkeit) für die Beteiligten.

Wie diese *logisch* meist inkompatiblen und zudem oft nur vage angebbaren Anforderungen in ein umfassendes *Konzept* sozialadäquater Sy-

stemgestaltung integriert werden können, ist Gegenstand weiterer Forschungen. *Steinmüller*

Angreifer

Teilgebiet: Informatik im Bankwesen

Eine Person, die einen *Angriff* auf ein *Informationssystem* vornimmt. *Martiny*

Angriff

Teilgebiet: Informatik im Bankwesen

Das Eindringen oder der Versuch des Eindringens in ein *Informationssystem*, um sich unberechtigt *Zugriff* auf Daten zu verschaffen und/oder diese zu ändern. Man unterscheidet aktive Angriffe und passive Angriffe. *Martiny*

Angriff, aktiver

Teilgebiet: Informatik im Bankwesen

Ein *Angriff*, der sich nicht auf die Einsicht von Daten beschränkt, sondern die Einspeisung von Daten beinhaltet. *Martiny*

Angriff, passiver

Teilgebiet: Informatik im Bankwesen

Ein *Angriff*, der keine Veränderung von Daten beinhaltet (z. B. das Abhören von Leitungen). *Martiny*

ANI

Teilgebiet: Telekommunikation

Automatic Number Identification

Siehe auch: CLI¹

Mit dem Anruf wird die *Nummer* des rufenden Teilnehmers übermittelt. Mit der Einführung von *ISDN* auch in Deutschland verfügbar. Die digitalen *Teilnehmer* im *ISDN* haben jedoch auch die Möglichkeit, die eigene Rufnummer zu unterdrücken (*Datenschutz*). Die Rufnummern-Kennung steht auch für alle *analogen* Teilnehmer an *ISDN*-Ortsämtern zur Verfügung. Die Freischaltung erfordert aus Datenschutzgründen jedoch einen entsprechenden Auftrag an die Telekom und ist mit dem Nachteil behaftet, daß eine dynamische Abschaltung für einzelne Anrufe zur *Zeit* noch nicht möglich ist. *Kuhn*

Animation

Teilgebiete: Computer Graphics, CAD und Bildverarbeitung; Simulation

animation

Synonym: Prozeßvisualisierung

Siehe auch: Trickfilmgraphik

Unter Computer-Animation versteht man die Technik der Erzeugung von bewegten oder belebten Bildern durch den Computer. Im Computer gespeicherte Objekte und deren Zustände und Veränderungen werden als eine Folge von Bil-

dem beschrieben, die auf unterschiedliche Art und Weise interpretiert werden können (zwei- bzw. dreidimensionale *Darstellung*, Farbe, Zusatzinformationen). Ein Einsatzschwerpunkt der Animation ist die *Visualisierung* von *Simulation*svorgängen, z.B. im Bereich der Planung und Auslegung von Fertigungssystemen oder der Fertigungssteuerung. Hierbei liegen die besonderen Vorteile der Animation in der Unterstützung bei der Entwicklung entsprechender *Simulationsmodelle* (Fehlersuche (→ *Debugging*), *Verifikation*, Validierung (→ *Validierung*²)), der Bereitstellung leicht verständlicher Analyse- und Auswertungsmöglichkeiten und der hohen Akzeptanz (→ *Akzeptanzproblem*) bei Präsentation der Simulationsergebnisse. Technik zur Wiedergabe der zeitabhängigen Veränderungen bei der Simulation von Prozessen. *Gorny*

Anklicken

Teilgebiet: Interaktive Systeme, Mensch-Maschine-Dialog

Eingabe, bei der mit der *Maus* der *Zeiger* auf dem *Bildschirm* (*Cursor*) auf einen *graphisch* dargestellten „Bedienknopf“ oder ein Feld eines *Menüs* gebracht und durch kurzes Drücken einer Maustaste eine Aktion des Rechners (→ *Datenverarbeitungsanlage, digitale*) ausgelöst wird. Durch „Doppelklick“, d.h. das zweimalige Drücken in kurzem zeitlichem Abstand, kann eine weitere Differenzierung - z.B. das Öffnen einer im Menü ausgewählten *Datei* - erfolgen. *Nowak*

Ankunftsrate → *Verkehrsquelle*

Anlagen-Konfiguration → *Konfiguration eines Datenverarbeitungssystems*

Anonymisierung

Teilgebiet: Datenschutz

Der Vorgang, bei dem der (direkte) Bezug zu einer benannten Person aus den personenbezogenen Daten eliminiert wird, wird als Anonymisierung bezeichnet. Anonyme Daten werden vor allem für *statistische* Zwecke (z.B. Markt- und Meinungsforschung) verwendet. Das *Bundesdatenschutzgesetz* ermöglicht in bestimmten Fällen die Weitergabe anonymisierter Daten an „Dritte“.

Es sind in der Datenschutzliteratur Fälle genannt worden, bei denen trotz Anonymisierung aus den „restlichen“ Merkmalen (→ *Attribut*) auf bestimmte Personen geschlossen wurde. *Schneider*

Anpaßbarkeit

Teilgebiet: Datenstrukturen und Datenoperationen
adaptability

Die Anpaßbarkeit (Adaptabilität) eines Programms bezeichnet den Grad seiner Anpassungsfähigkeit an verschiedene Benutzerbedürfnisse. Sie ist um so höher, je geringer Anzahl und Umfang der bei einer Anpassung erforderlichen Programmeeingriffe (Hinzufügen, Entfernen und Ändern von Programmteilen) sind. Eine hohe Adaptabilität strebt man dann an, wenn innerhalb eines zugrundeliegenden Problemtyps eine breite Palette von Benutzerbedürfnissen mit einem Programm abzudecken ist. Programme, die diese Forderung in hohem Maße erfüllen, bezeichnet man als *Standardprogramme*.

Eine hohe Adaptabilität läßt sich durch die Anwendung geeigneter Entwurfstechniken, wie z.B. die *modulare Programmierung*, erreichen. Von der Adaptabilität ist die *Portabilität* zu unterscheiden. Hohe Adaptabilität vergrößert die Portabilität, da sie den Benutzer in die Lage versetzt, Programmeigenschaften zu eliminieren, welche mit den Komponenten eines Computersystems (→ *Datenverarbeitungssystem*), wie z.B. Arbeitsspeichergröße (→ *Hauptspeicher*), unverträglich sind. *Gehring*

Anpassungsschicht → *Kommunikationssystem, offenes*

Anreicherungstyp → *MOS-Technik*

Ansatz, algorithmen-orientierter

Teilgebiet: Computergestützter Unterricht und Pädagogik

Methoden des algorithmischen Problemlösens im Informatikunterricht einzuüben und anzuwenden, sind zentrales Anliegen dieses Ansatzes. Probleme aus der Erlebniswelt des Schülers werden systematisch strukturiert und schrittweise über die Problemanalyse (→ *Problemlösung*) gelöst: Das *Problem* wird erkannt, ein Lösungsplan wird entworfen, das Problem wird gelöst, die Lösung wird strukturiert beschrieben (→ *Algorithmus*), in eine *Programmiersprache* übertragen und auf dem Rechner getestet. Im allgemeinen kann der Anwendungszusammenhang (→ *Anwender*) für das entstehende Programm in seiner Vielschichtigkeit im Unterricht (→ *Unterricht, computerunterstützter*) nur begrenzt behandelt werden. *Gorny*

Ansatz, anwendungs-orientierter

Teilgebiete: Computer in der Schule; Computergestützter Unterricht und Pädagogik

Nach diesem Ansatz sollen sich Schüler mit Anwendungen und Auswirkungen der Informatik- und Informationstechnik in ihrem Umfeld bekannt machen. Ausgehend von typischen Anwendungszusammenhängen und praktischen Fragestellungen

werden die dahinterliegenden Problemstellungen durch Verallgemeinerung und *Abstraktion* mit Methoden der Datenstrukturierung und der Algorithmenkonstruktion formuliert. Die Problembearbeitung kann mit Hilfe von Standardsoftware-Paketen und deren Beschreibungssprachen oder mit spezieller *Lehr- und Lernsoftware* erfolgen.

Gorny

Ansatz, hardware-orientierter

Teilgebiete: Computer in der Schule; Computergestützter Unterricht und Pädagogik

Nach diesem Ansatz soll ein Einblick in die mathematisch-physikalisch-technische Funktionsweise des Rechners (→ *Datenverarbeitungsanlage, digitale*) vermittelt werden. In mehreren Entwurfsebenen wird das informationstechnische Gerät (→ *Peripheriegerät*) synthetisch entworfen und mit Hilfe der *Schaltalgebra* aus *elektronischen* Elementarbausteinen auf den elektronischen Schaltnetz- und Schaltwerkebenen nachgebaut. So entsteht ein theoretischer Modellcomputer, der mit nur geringem Befehlssatz (→ *Befehlsvorrat*) nur maschinennah (→ *Maschinenabhängigkeit*) zu programmieren ist. Praxisnahe Anwendungen lassen sich mit ihm nicht realisieren und damit auch nicht im Unterricht (→ *Unterricht, computerunterstützter*) deren Folgen und Wirkungen diskutieren.

Gorny

Ansatz, informatischer

Teilgebiete: Computer in der Schule; Computergestützter Unterricht und Pädagogik

Der informatische Ansatz (insbesondere in der Sekundarstufe II der allgemeinbildenden Schulen) wurde zunächst als theoretischer Ansatz bezeichnet, der einen strukturellen Zugang zur Informationstechnik *sucht*, mit dem Ziel, den Algorithmenbegriff, die Eigenschaften formaler Sprachen und die *Berechenbarkeit* im Unterricht (→ *Unterricht, computerunterstützter*) zu untersuchen (siehe: *Fachdidaktik der Informatik*). Die Erweiterung dieses Ansatzes soll dem Schüler eine „informatische Weltanschauung“ vermitteln: Die informatische Interpretation von natur- und ingenieurwissenschaftlichen sowie betriebswissenschaftlichen Problemen ebenso wie die Untersuchung sprachwissenschaftlicher Fragestellungen, der *Wissenserwerbs*, der Wissensverarbeitung und der Methoden des Schlußfolgerns bilden in diesem Ansatz eine notwendige Voraussetzung für die angemessene *Nutzung* der Informationstechnik.

Gorny

Ansatz, integrativer

Teilgebiete: Computer in der Schule; Computergestützter Unterricht und Pädagogik

Sehr nahe dem anwendungsorientierten Ansatz liegt dieses didaktische *Konzept* (siehe: Konzept, didaktisches) für die Informatische (Informations- und Kommunikationstechnische) Grundbildung (siehe: Grundbildung, informatische). Der Unterschied zum anwendungsorientierten Ansatz besteht in dem Verzicht auf ein spezielles Fach („Informatik“, „Informations- (und kommunikations-) technische Grundbildung“ o.ä.) oder auf ein „Leitfach“, das die Hauptlast der Einführung in die Informationstechnik zu tragen hat. Stattdessen werden die unterschiedlichen Aspekte der Informatik in verschiedene Fächer integriert, zum Beispiel die algorithmischen und formallogischen Aspekte in die Mathematik, die sozialen und anwendungsorientierten in die Fächer „Sozialkunde“ und „Arbeit/Wirtschaft“, die Betrachtung der *Hardware* in die Physik, Fragen der formalen Sprachen in den Sprachunterricht (Deutsch und Fremdsprachen), Konzepte der *Modellierung* (siehe: Modell) und *Simulation* von komplexen Systemen in die Biologie und Physik, die Probleme der Kommunikation, der Formalisierung von Information und der *Darstellung* von Information in unterschiedlicher *Notation* etwa in die Fächer Kunst, Musik und Deutsch.

Gorny

Ansatz, kultur-orientierter

Teilgebiete: Computer in der Schule; Computergestützter Unterricht und Pädagogik

Dieser Ansatz hat zum Leitziel die Befähigung des Menschen zur sinnvollen und methodischen *Nutzung* der Informations- und Kommunikationstechnik in allen Lebensbereichen. Dem Schüler steht bei diesem Vorgehen der *Computer in der Schule* als universelles interaktives (→ *Interaktion*) Lernwerkzeug in einer computerunterstützten Lernumgebung (siehe: Lernumgebung, computerunterstützte (→ *computergestützt*)) zur Verfügung.

Das Lernziel ist die Befähigung zum systematischen und kritischen Umgang mit der Informations- und Kommunikationstechnik. Dieser Umgang wird als eine der grundlegenden „Kulturtechniken“ verstanden, die jeden Schüler für einen Zugang zu Informationsquellen (→ *Datenquelle*) und zur Nutzung von Computern „zum Zeichnen, Schreiben, Komponieren, Entdecken, Berechnen, *Analysieren*, Rollen-Spielen und zum *Zugriff* auf Informationssysteme“ (Regierung von Ontario 1987) befähigen soll. Organisatorisch erfordert der kultur-orientierte Ansatz eine weitgehende Reform der Unterrichtsinhalte und -methoden in allen Schulfächern und eine Ausstattung der Schulen mit Computern in allen

Klassenzimmern, die einen ständigen, selbständigen und selbstverständlichen Zugang zu den Software-Werkzeugen und Informationsquellen ermöglichen. *Gorny*

Anschalteinheit → *Datenübertragungseinrichtung*

Anschalteinheit für Kartentelefone → *AEK*

Anschläge in der Textverarbeitung

Teilgebiet: Textverarbeitung und Büroautomation
keystroke

Anzahl mittels einer Schreibmaschinentastatur geschriebener (in der DV: eingegebener) Zeichen (*Buchstaben* und *Leerschläge*) pro Text- (*Zeile*, Seite) oder Zeiteinheit. *Ehlers*

Anschlußelektronik → *Heimcomputer*

Anschlußfahne → *DIL*

Anschlußfläche → *Bonden*

Anschlußfleck → *DIL*

Anschlußstelle → *Schnittstelle*

Anschriftenleser

Teilgebiet: Rechnerperipherie
address reader

Anschriftenleser sind spezielle OCR-Systeme, die in der *Automatisierung* des Postwesens aller Industrieländer eine wichtige Rolle spielen. Aufgabe des Anschriftenlesers ist, die Adresse auf dem Poststück zu finden, zu *lesen* und richtig zu interpretieren. Gelesen wird die gesamte Anschrift: Orts- und Straßename mit Postleitzahl, sofern vorhanden Gebäude- und Stockwerks- und Firmenbezeichnung. Mit diesen Informationen ist die *Sortierung* bis in die Tasche des Briefträgers möglich. *Schürmann*

ANSI

Teilgebiet: Allgemeines
Abkürzung von: American National Standards Institut

Nationaler Normenausschuß der USA. Er entspricht dem *DIN* in der Bundesrepublik Deutschland. *Eckert*

Anspruchsklasse

Teilgebiet: Qualitätsmanagement

Kategorie oder Rang unterschiedlicher Qualitätsforderungen an Einheiten für den gleichen funktionellen Gebrauch. Anmerkungen:

1. Die Anspruchsklasse spiegelt einen geplanten oder anerkannten Unterschied in der *Qualitätsforderung* wider. Die Betonung

liegt auf der Beziehung zwischen funktionellem Gebrauch und *Kosten*.

2. Eine Einheit hoher Anspruchsklasse (z.B. ein Luxushotel) kann von nicht zufriedenstellender *Qualität* sein, und umgekehrt.
3. Wo Anspruchsklassen *numerisch* gekennzeichnet sind, ist es üblich, daß die höchste Anspruchsklasse als 1 bezeichnet ist, wobei sich die niedrigeren Anspruchsklassen auf 2, 3, 4 usw. erstrecken. Wo Anspruchsklassen mit einer Punkteskala bezeichnet sind, etwa durch eine Anzahl von Sternen, hat die niedrigste Anspruchsklasse üblicherweise die wenigsten Punkte oder Sterne. (nach *DIN EN ISO 8402*, Ausgabe August 1995)

Möller

Ansteuerung

Teilgebiet: Rechnerarchitektur

Aufschalten von *Steuerinformation* an gewisse Eingänge (Steuerereingänge) einer Hardware-Funktionseinheit zur Steuerung ihrer Funktionsweise. *Giloi*

Anstiegszeit

Teilgebiet: Allgemeines
rise time

Siehe auch: Schaltzeit

Die Anstiegszeit eines Signales ist die Zeit, die vergeht, um von 10% auf 90% der Maximalamplitude eines Signales zu gelangen. *Hoffmann, R.*

Anstoßmultiplikator → *Multiplikatoranalyse*

Answer to reset

Teilgebiet: Informatik im Bankwesen
Abkürzung: ATR

Antwort einer *Chipkarte* nach dem Einschalten bzw. auf einen *Rücksetzimpuls*. Im ATR werden verfügbare Chipkartenprotokolle und Übertragungsparameter angezeigt. *Martiny*

Antialiasing

Teilgebiet: Desktop Publishing
antialiasing

Werden im *Belichter* einer Satzanlage oder auf dem Videobildschirm Zeichen durch diskrete Einheiten zusammengesetzt, ergeben sich – je nach der Auflösung des Visualisierungsgerätes – bei Rundungen, Bögen oder Schrägen treppenstufenartige Verzeichnungen. Durch Antialiasing-Routinen und andere Randglättungsverfahren lassen sich – durch geeignete *Filter* – diese Stufen abschwächen, die scharfen Kanten wirken weicher und etwas verschwommen. *Ehlers*

antikanonische Ableitung → *Rechtsableitung*

Antiqua

Teilgebiet: Textverarbeitung und Büroautomation

Roman type

Antiqua ist eine Schriftgattung mit runden Formen, deren *Buchstaben* innerhalb eines Wortes nicht gebunden/verbunden sind. Die Großbuchstaben wurden aus den römischen Kapitalbuchstaben entwickelt.

Ehlers

Antivalenz

Teilgebiet: Logik

non-equivalence

Synonym: exklusives Oder

Eine zweistellige *boolesche Funktion* mit der Funktionstabelle

a	b	$a \neq b$
0	0	0
0	1	1
1	0	1
1	1	0

Als Funktionssymbole sind gebräuchlich: $a \neq b$, $a \oplus b$.

Reusch

ANT Nachrichtentechnik GmbH

Teilgebiet: Allgemeines

Die ANT Nachrichtentechnik GmbH, Backnang, arbeitet auf allen Gebieten der leitungsgebundenen und drahtlosen *Nachrichtenübertragung*. Das Arbeitsprogramm umfaßt die Planung, Entwicklung, Produktion, den Vertrieb und die Inbetriebnahme und *Wartung* von Nachrichtenanlagen und -netzen im In- und Ausland.

Seit 1. Januar 1988 liegen die Geschäftsanteile der ANT Nachrichtentechnik GmbH bei der Robert Bosch GmbH, Stuttgart (82%), und der Allianz Versicherungs-AG, München (18%).

Die Gesellschaft ist auf den Gebieten *Multiplex-technik*, *Nachrichtensatellitentechnik*, *Glasfaserkabel* und vollständige Fernmeldekabelanlagen, Richtfunktechnik, *GSM* Funksysteme und eine Reihe spezieller *Kommunikationssysteme* tätig.

Die ANT Nachrichtentechnik GmbH beschäftigt 4800 Mitarbeiter in den Standorten Backnang, Schwäbisch Hall und Offenburg sowie in ihren neun inländischen Vertriebsniederlassungen und Stützpunkten.

Für das Auslandsgeschäft unterhält ANT in den USA eine Tochtergesellschaft.

Mit Wirkung vom 1. Januar 1988 an wurden die Geschäftsanteile der Tedix GmbH, Heidelberg, in das Unternehmen eingebracht. Damit einherging die Erhöhung des Stammkapitals auf DM (→ *Wartezustand*) 240 Mio.

Rund ein Fünftel der Mitarbeiter ist im Bereich Forschung und Entwicklung tätig.

Deuse

Antwortanalyse → *Unterricht, programmierter; Autorensprache*

Antwortanalyseverfahren

Teilgebiet: Computergestützter Unterricht und Pädagogik

In einem Lehrprogramm (→ *Lehrprogramme, Klassifikation von*) wird der (individuelle) Weg eines jeden Lernenden durch die Antworten auf die programmierten Fragen ausgewählt. Die Möglichkeiten zur *Analyse* der Schülerantworten sind daher wichtige Bestandteile aller *Autorensprachen*.

Ein häufig benutztes einfaches Antwortanalyseverfahren stellt die Mehrfach-Auswahl-Antwort (→ *Multiple-Choice*) dar. Dabei werden mehrere mögliche Antworten, einzeln bezeichnet durch einen Buchstaben oder eine Ziffer, vorgegeben; der Lernende wählt die passende Antwort durch Angabe des Buchstabens, bzw. der Ziffer aus.

Oft werden *Schlagwörter* (keywords) aus dem Text extrahiert und zur Beurteilung der Antwort verwendet. Bei der Freiantwortanalyse werden ganze *Sätze* zunächst syntaktisch (→ *Syntax von Programmiersprachen*), dann auf ihren Sinn hin analysiert. Da dieses *Verfahren* noch einige Schwierigkeiten bereitet, wird es nur in speziellen Gebieten (z.B. im Rahmen einer *Fachsprache*) verwendet.

Schneider

Antwortauswertung → *Autorensprache*

Antwortextraktion

Teilgebiet: Kognitive Methoden – Künstliche Intelligenz

Algorithmisches *Verfahren* zur Gewinnung einer prädikatenlogischen (→ *Prädikatenlogik*) Antwort auf eine vorgelegte Frage, auf der Grundlage des *Widerlegungsgraphen* nach dem Resolutionsprinzip (→ *Resolventenprinzip*).

Anwendung in *Frage-Antwort-Systemen*.

Wernicke

Antwortprogramm auf ein Unterbrechungssignal

Teilgebiet: Prozedatenverarbeitung

Unter Antwortprogramm auf ein Unterbrechungssignal versteht man dasjenige Programmstück im Rechner, das nach der Wahrnehmung eines *Unterbrechungssignals* durch den *Prozessor* und nach der Identifizierung gezielt als Antwort auf das zugehörige externe Vorkommnis im *technischen Prozeß* ausgeführt wird.

Lauber

Antwortverhalten

Teilgebiet: Betrieb von Rechensystemen
time sharing performance

Reaktion des *DV-Systems* im *interaktiven Betrieb* (z.B. *Terminal*) auf Eingaben des Benutzers. Charakteristische Größe für das Antwortverhalten ist die Antwortzeit (\rightarrow *Antwortzeit*²) oder Beantwortungszeit, die Zeitspanne zwischen dem Ende einer Aufgabenstellung und dem Vorliegen der vollständigen Antwort darauf (*DIN* 44300). Angestrebt werden möglichst kurze Antwortzeiten. Ferner soll die Antwortzeit für den Benutzer kalkulierbar sein, d.h. die *Varianz* der Antwortzeit für die Ausführung eines bestimmten *Kommandos* oder einer Gruppe ähnlicher *Kommandos* soll klein sein. Datenbankanwendungen moderner Transaktionssysteme (\rightarrow *Transaktion*) verlangen Antwortzeiten von 0,5 sec. bis 2 sec. *Haupt*

Antwortzeit¹

Teilgebiet: Betriebssysteme
*response time*¹

Bei einer Instanz die Zeitspanne zwischen dem Zeitpunkt, zu dem sie die Erteilung eines Auftrages an eine andere Instanz beendet, und dem Zeitpunkt, zu dem bei der auftraggebenden Instanz die Übergabe des Ergebnisses der Auftragsbearbeitung oder einer Mitteilung darüber beginnt. Die Antwortzeit hängt vom Umfang der zu erbringenden Datenverarbeitungsleistung ab. Häufig ist die auftraggebende Instanz eine *Benutzerstation*.

Dieser Festlegung liegt folgende Modellvorstellung (\rightarrow Modell, (3.)) zugrunde. Eine Zeitspanne wird durch einen Anfangs- und einen Endzeitpunkt festgelegt. Im allgemeinen sind die Zeitpunkte zwischen dem Anfangs- und Endzeitpunkt nicht von Interesse. Wo jedoch von dem Kontinuum zwischen Anfangs- und (damit nicht identischem) Endzeitpunkt die Rede sein soll, nennt man dies einen Zeitabschnitt oder ein Zeitintervall; dieser bzw. dieses hat eine Dauer.

Ein Ereignis wird gedacht als derjenige Grenzfall eines Vorganges, der die Dauer Null hat. Das Eintreten eines Ereignisses definiert somit einen Zeitpunkt. Ersetzt man in der Festlegung einer Zeitspanne die kennzeichnenden Zeitpunkte durch das sie jeweils definierende Anfangs- bzw. Endereignis, so ist es außerdem notwendig, den Ort des Eintretens für jedes der kennzeichnenden Ereignisse zu nennen. Als Ort des Eintretens fungieren häufig *Instanzen* (nach *DIN* 44 300 T7). *Fleischhauer; Guse; Rouette*

Antwortzeit²

Teilgebiet: Prozedatenverarbeitung

response time

Die Antwortzeit ist die Zeitspanne zwischen der Erzeugung eines externen *Unterbrechungssignals* im *technischen Prozeß* und der Ausführung des letzten Befehls des zugehörigen *Unterbrechungs-Antwortprogramms* im *Prozeßrechner*. Sie wird durch die Eigenschaften des *Betriebssystems* des Rechners (\rightarrow *Datenverarbeitungsanlage, digitale*) und durch die momentane Last beeinflusst. Das Bild auf S. 42 gibt eine Übersicht über die Zeitintervalle, aus denen sich die Antwortzeit zusammensetzt (*Ausführungszeit, Durchlaufzeit, Latenzzeit* (\rightarrow *Latenzzeit*¹), *Erkennungszeit*). Abweichend von der obigen *Definition* wird in der Literatur häufig *Reaktionszeit* für Antwortzeit verwendet. *Lauber*

Antwortzeit bei interaktiven Systemen

Teilgebiet: Betriebssysteme

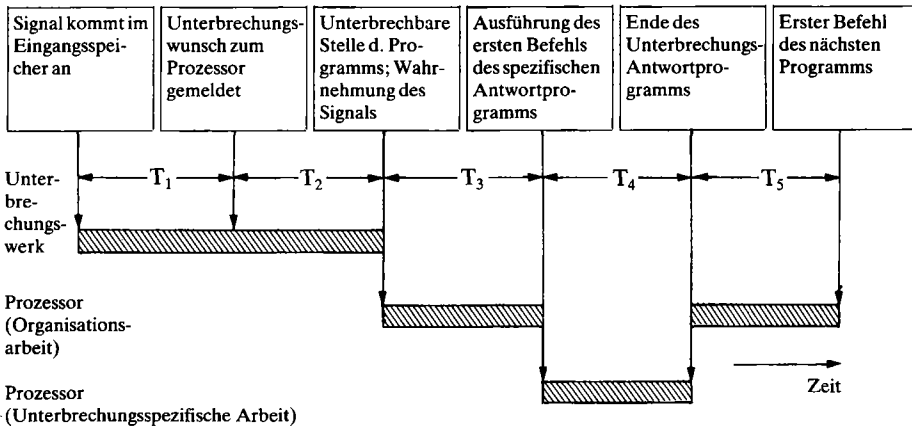
Synonym: Reaktionszeit bei interaktiven Systemen

Die Zeit zwischen dem Ende einer *Benutzereingabe* und dem Moment, in dem die darauffolgende *Ausgabe* vom Benutzer wahrgenommen werden kann. Diese *Definition*, die außer *Rechenzeiten* auch Eigenschaften des Ausgabemediums (\rightarrow *Ausgabeeinheit*) in Betracht zieht, ist für *interaktive Systeme* (\rightarrow *Echtzeitsystem*) angemessen. Beispiel: bei einigen *Ausgabegeräten* ist eine Ausgabe nicht lesbar, bevor eine ganze *Zeile* fertig geschrieben ist; in diesem Fall erstreckt sich die Ausgabezeit bis zu dem Moment, da die Zeile vollständig geschrieben ist.

Der Benutzer eines *interaktiven Systems* erwartet, daß die Antwortzeit (\rightarrow *Antwortzeit*²) erstens voraussagbar ist, zweitens der zu verarbeitenden Aufgabe entspricht. Zum ersten *Punkt* zeigen viele *Experimente*, daß Benutzer durchschnittlich längere Antwortzeiten akzeptieren, falls die *Varianz* der Antwortzeiten klein ist. Zum zweiten *Punkt*: in jedem *Dialog* treten triviale Eingaben (*Befehle*) auf, wie z.B. „nächste Seite zeigen“, die wenig *Verarbeitung* verlangen. Die Antwort auf triviale Eingaben sollte bei interaktiven Systemen augenblicklich erfolgen, die Antwortzeit also an der Grenze des zeitlichen Auflösungsvermögens des Menschen liegen. Eine Zehntelsekunde erfüllt diese Forderung. Leider liegen bei vielen heutigen *Teilnehmersystemen* Antwortzeiten immer über einigen Sekunden.

Im Dialogbetrieb liegt sie in der Größenordnung von „augenblicklich“ bis zu wenigen Sekunden, im *Echtzeitbetrieb* (\rightarrow *Echtzeitverarbeitung*) vielfach unter $\frac{1}{10}$ Sekunden. Die Antwortzeit beinhaltet nicht immer die vollständige Abarbeitung des durch den Eingabevorgang spezifizier-

Anweisung



Definitionen:

T_1 = Durchlaßzeit

T_2 = Latenzzeit

T_3 = Erkennungszeit

T_4 = Ausführungszeit
(umfaßt nur den Ablauf des unterbrechungsspezifischen Antwortprogramms. Identifizierung der Unterbrechungsursache und Bereitstellung der Betriebsmittel sind zu T_3 zu rechnen)

T_5 = Rückkehrzeit

$T_1 + T_2$ = Wartezeit

$T_1 + T_2 + T_3$ = Reaktionszeit

$T_3 + T_4 + T_5$ = Unterbrechungszeit

$T_1 + T_2 + T_3 + T_4$ = Antwortzeit

$T_3 + T_5$ = Organisationszeit

$T_1 + T_2 + T_3 + T_4 + T_5$ = Gesamtzeit

ten Auftrags. Dieser kann ggf. einem *Hintergrundstapelbetrieb* übergeben werden. Es ist lediglich die Zeit, für die die *Datenstation* blockiert ist und keine neuen Eingabewerte annehmen kann. Das Analogon zur Antwortzeit ist im *Stapelbetrieb* die *Umschlagszeit*. *Endres; Nievergelt*

Anweisung

Teilgebiet: Programmierung

Nach den Regeln einer beliebigen Sprache festgelegte syntaktische (\rightarrow *Syntax von Programmiersprachen*) Einheit, die in gegebenem oder unterstelltem Zusammenhang wie auch im Sinne dieser Sprache eine Arbeitsvorschrift ist. Im allgemeinen bildet eine bestimmte Aufgabenstellung diesen Zusammenhang; aus ihr ergibt sich die Arbeitsvorschrift. Die Anweisung richtet sich an diejenige Person oder Funktionseinheit, welche die Arbeit ausführen soll. In der Datenverarbeitung werden Anweisungen meistens in einer *Programmiersprache*, *Betriebsprache* oder *Kommandosprache* formuliert. Anweisungen können nach Art der Arbeitsvorschrift *klassifiziert* werden. Wichtige Klassen sind z.B.

- bedingte Anweisung mit Verzweigungsanweisung, Wiederholungsanweisung,

- unbedingte Anweisung mit Zuweisung, Sprunganweisung (\rightarrow *Sprungbefehl*), Eingabe- u. Ausgabeanweisung, Transportanweisung.

Manche Programmiersprachen lassen es zu, daß eine Anweisung Teile enthält, die selbst Anweisungen oder *Vereinbarungen* sind. Das gilt nicht für elementare Anweisungen (nach *DIN 44 300 T4*). *Fleischhauer; Guse; Klar; Rouette*

Anweisung, bedingte

Teilgebiete: Programmierung; Rechnerorganisation
conditional statement

Eine Anweisung, die eine Bedingung enthält. In Abhängigkeit vom Ergebnis der *Prüfung* dieser Bedingung wird der *Programmablauf* zu Teilen der bedingten Anweisung oder zu anderen Anweisungen verzweigt.

Fleischhauer; Guse; Klar; Rouette

Anweisung, bewachte

Teilgebiet: Programmierung
guarded command; guarded statement

Die bewachte Anweisung ist eine verallgemeinerte Form der bedingten Anweisung. Sie wird

üblicherweise geschrieben in der Form

if $B_1 \rightarrow S_1 \square B_2 \rightarrow S_2 \square \dots \square B_n \rightarrow S_n$ fi

Die *Booleschen* Ausdrücke B_i heißen Wächter (guards), die S_i sind Anweisungen. Die Auswertung erfolgt üblicherweise so, daß irgendein Wächter B_i gesucht wird, der true (\rightarrow Wahrheitswert) liefert, und dann die zugehörige Anweisung S_i ausgeführt wird. Das Durchmustern der Wächter erfolgt in beliebiger Reihenfolge; die Auswertung ist also *nichtdeterministisch*. Ergibt kein Wächter true, dann ist die gesamte Anweisung undefiniert.

In der Literatur finden sich auch Varianten dieser *Semantik*. So kann man z.B. fordern, daß die gesamte Anweisung undefiniert ist, sobald wenigstens einer der Wächter undefiniert ist („böswilliger Nichtdeterminismus“). Alternativ dazu kann man auch festlegen, daß undefinierte Wächter ignoriert werden, sobald wenigstens ein Wächter true liefert („gutwilliger Nichtdeterminismus“).

Die Idee der bewachten Anweisung wird auch mit dem Konzept der *Iteration* verbunden zur bewachten Wiederholungsanweisung

do $B_1 \rightarrow S_1 \square B_2 \rightarrow S_2 \square \dots \square B_n \rightarrow S_n$ od,

die so lange auszuführen ist, bis kein B_i mehr true liefert. Pepper

Anweisung, zusammengesetzte

Teilgebiet: Programmierung
compound statement

Synonym: Verbundanweisung

Eine Anweisung in einem in einer *höheren Programmiersprache* formulierten Programm, die textuell selbst wieder Anweisungen enthält (beispielsweise einen Block (\rightarrow *Block¹*), eine *Laufanweisung*, eine bedingte Anweisung).

Hoffmann, H.-J.

Anweisungstest \rightarrow *Testmethode*

Anwender

Teilgebiet: Informatik in der Betriebswirtschaft
Eine Organisation bzw. Institution, die Rechner-systeme zur Erfüllung von *Datenverarbeitungsaufgaben* bzw. zur Unterstützung von *Informati-onsverarbeitungsprozessen* einsetzt. Hierbei ist es unerheblich, welchen Umfang die Rechner-Anwendung hat.

Ergänzung: Anwender-Institutionen bzw. -Organisationen setzen Rechner häufig für viele unterschiedliche Zwecke ein. Der Begriff „Anwender“ wird hier bewußt nicht auf eine einzelne Person oder auf eine Personengruppe bezogen, die Rechner für ihre speziellen Aufgaben einsetzt. In letzterem Falle wird der Begriff „Benutzer“ verwendet. Die Begriffe Anwender und Benutzer wer-

den hier nicht synonym verwendet: Bei einem Anwender gibt es üblicherweise viele verschiedene Benutzer. Seibt

Anwenderberatung \rightarrow *Benutzerberatung*

Anwenderbetreuung \rightarrow *Datenverarbeitungsorganisation*

Anwenderkoordinaten \rightarrow *Benutzerkoordinaten*

Anwender-Organisation \rightarrow *Benutzer-Organisation*

Anwenderprogramm \rightarrow *Anwendungsprogramm*

Anwenderprozeß

Teilgebiet: Prozeßdatenverarbeitung

Synonym: Benutzerprozeß

Rechenprozeß, der vom Anwender (Benutzer) eingerichtet ist. Lauber

Anwendung

Teilgebiet: Informatik in der Betriebswirtschaft
application; application system

Synonym: Applikation

Anwendungsprogramme zusammen mit den von diesen Programmen zu verarbeitenden Daten und den hinter den Programmen stehenden ablafor-ganisatorischen Regelungen, die notwendig sind, um die Anwendungsprogramme in einer realen Umgebung (*Betrieb*, Behörde usw.) zu implementieren (\rightarrow *Implementierung*) und zur Wirkung zu bringen.

Ergänzung: In der Praxis der automatischen Datenverarbeitung werden auch die für bestimmte Funktionsbereiche geschaffenen Programme zusammenfassend als „Anwendungen“ bezeichnet. So wird beispielsweise von „Anwendungen im Bereich des Marketings“ gesprochen. Man versteht darunter Programme, die die Erfüllung der *Informationsverarbeitungs*aufgaben im Marketing unterstützen.

Der Ausdruck „Anwendung“ wird zwar in der Praxis häufig, aber mehrdeutig benutzt. Seibt

Anwendung, interaktive

Teilgebiet: Informatik in der Betriebswirtschaft
interactive application

Eine Anwendung, bei der die Betriebsart (\rightarrow *Re-chensystem, Nutzungsformen eines*) des Dialogbetriebs praktiziert wird.

Ergänzung: In Ergänzung zum Begriff „*Dialog betrieb*“, der lediglich auf die Art des durchgeführten Rechnerbetriebs (\rightarrow *Betriebsart eines Rechensystems*) abhebt, umfaßt der Begriff „*interaktive Anwendung*“ auch alle Programme, ablauforganisatorischen, evtl. gerätetechnischen

Anwendung, kryptographische

(→ *Peripheriegerät*) Voraussetzungen bzw. Regelungen, die erforderlich sind, um wirksame Rechner-Unterstützung (→ *computergestützt*) für einen oder mehrere bestimmte *Informationsverarbeitungsprozesse* in bestimmter Umgebung herbeizuführen. Seibt

Anwendung, kryptographische

Teilgebiet: Kryptographie

crypto application

Synonym: Kryptoanwendungen

Zu unterscheiden sind:

1. Einsatz zum Schutz von Daten in Computern und *Peripheriegeräten*. Hierbei muß die *Computersicherheit* berücksichtigt werden und
2. Einsatz zum Schutz von *Nachrichten* auf Übertragungswegen. Man unterscheidet die Einzelkanalverschlüsselung (*Verschlüsselung* eines *Kanals* für Sprache, Daten, Fernschreiben (→ *Telex*), *Faksimile* etc.) und die Bündelverschlüsselung (z.B. Verschlüsselung eines 2,048 / 448 PCM (→ *Pulscode-modulation*) Bündels). Die Verschlüsselung kann in Form der *End-zu-End-Verschlüsselung* oder der abschnittswisen Verschlüsselung eingesetzt werden.

Angewandt wird die Verschlüsselung in

- a) Leitungsgebundenen Netzen
 - Einsatz auf festgeschalteten Leitungen oder in leitungsvermittelten Netzen, wobei die Verschlüsselung im allgemeinen prozedurunabhängig (transparent) ist.
 - Einsatz in Paketvermittlungsnetzen mit prozedurabhängiger End-zu-End-Verschlüsselung oder prozedurabhängiger abschnittsweiser Verschlüsselung.
- b) Funknetzen.

Deuse

Anwendungsanalyse

Teilgebiet: Informatik in der Betriebswirtschaft
application analysis

Synonym: Analyse des Anwendungsgebiets

Analyse, durch die festgestellt wird,

- a) ob ein bestimmter Funktionsbereich (= Bereich von *Informationsverarbeitungsaufgaben*) mit Hilfe von Rechnersystemen unterstützt werden kann (*Durchführbarkeitsstudie*, technische Aspekte);
- b) ob diese Rechner-Unterstützung (→ *computergestützt*) voraussichtlich zu einer Verbesserung der Wirksamkeit und Wirtschaftlichkeit der Aufgabenerfüllung in diesem Bereich führen wird (*Durchführbarkeitsstudie*, ökonomische Aspekte).

Ergänzung: Anwendungsanalysen werden in der Praxis üblicherweise von Benutzern und *Datenverarbeitungsspezialisten* gemeinsam durchgeführt. Seibt

Anwendungsanbieter

Teilgebiet: Informatik im Bankwesen

Eine Organisation, die eine Anwendung einer Karte und die damit verbundene *Dienstleistung* anbietet und verwaltet. Der Anwendungsanbieter braucht nicht selbst *Kartenherausgeber* zu sein. Martiny

Anwendungsbereitstellung, fachliche

Teilgebiet: Programmierungstechnik

Gesamtheit aller Tätigkeiten, die seitens des Anwenders zur Einsatzplanung, Abnahme, Vorbereitung des Anwendungsumfelds, Datenbereitstellung, Vergabe von Anwendungsparametern und zur Einführung/Schulung erforderlich sind. Anmerkung: Diese Tätigkeiten werden häufig von einer Anwenderbetreuungsstelle wahrgenommen. Fritsche

Anwendungsbereitstellung, technische

Teilgebiet: Programmierungstechnik

Gesamtheit aller Tätigkeiten, die seitens des Anwenders zur Installation, Anpassung/Customizing, Korrektur-Einbringung, zum *Konfigurations-Management* und zur Archivierung erforderlich sind.

Anmerkungen:

1. Die Korrekturen selbst werden vom Herstellungssystem geliefert.
2. Diese Tätigkeiten werden häufig von einer Anwenderbetreuungsstelle wahrgenommen. Fritsche

Anwendungsbezeichner

Teilgebiet: Informatik im Bankwesen

Application Identifier, AID

Ein in der *Chipkarte* gespeichertes Datum, das eine Chipkartenanwendung eindeutig bezeichnet. AIDs werden unter Hoheit von *ISO* vergeben, um eine Eindeutigkeit sicherzustellen. Martiny

Anwendungsgenerator

Teilgebiet: Programmierung

application generator

Synonym: Programmiersprache der 4. Generation

Programmiersystem (*Generator*) zur effizienten Erstellung bestimmter Klassen von *Anwendungsprogrammen*. Die Einfachheit der Programmierung ergibt sich daraus, daß oft eine bestimmte Anwendungslogik fest vorgegeben ist, z.B. Be-

richerstellung oder Dateifortschreibung. In Verbindung mit Datenlexika (→ *Datenlexikon*), symbolischer Testausführung (→ *Testmethode*) und anderen Endbenutzer-Hilfen senken sie ganz erheblich die notwendigen fachlichen Vorkenntnisse für die Erstellung einfacher Programme.

Endres

Anwendungsmodell

Teilgebiet: Grundlagen von Informationssystemen

Modell, das sich auf einen Weltausschnitt bezieht, der *Gegenstandsbereich* einer Software-Anwendung ist.

Barkow; Hesse; Kittlaus; Scheschonk; von Braun

anwendungsorientiertes Protokoll → *Protokollhierarchie*

Anwendungsprogramm

Teilgebiet: Informatik in der Betriebswirtschaft *application program; user program*

Synonyme: Anwenderprogramm; Benutzerprogramm

Ein Programm zur Erfüllung einer oder mehrerer spezieller *Datenverarbeitungsaufgaben* von einem oder mehreren Benutzern oder von einer oder mehreren organisatorischen Einheiten.

Ergänzung: Im Gegensatz zu den vom *Hersteller* zusammen mit der *Hardware* gelieferten *Systemprogrammen* (*Betriebssystem*) bezeichnet man die in der Mehrzahl von den *Datenverarbeitungsspezialisten* des *Anwenders* für die speziellen *Datenverarbeitungsaufgaben* der Benutzer erstellten Programme als *Anwendungs- bzw. Anwenderprogramme* (→ *Anwendungssoftware*). Der *Begriff* ist allerdings nicht eindeutig festgelegt. Anwenderprogramme können auch von den *Datenverarbeitungsspezialisten* der Hersteller oder der *Software-Häuser* (→ *Software-Haus*) entwickelt worden sein. Dies geschieht sogar häufig für *Datenverarbeitungsaufgaben*, die in gleicher oder ähnlicher Form bei vielen Anwendern auftreten (mehrfach verwendbare bzw. *Standardanwendungssoftware* (→ *Standardanwendungs-Software*)). Zukünftig werden durch starke Verbreitung *benutzerorientierter* (→ *Benutzerorientierung*) *Programmiersprachen* immer mehr Benutzer und immer mehr Anwenderbereiche dazu übergehen, ihre eigenen Anwenderprogramme zu erstellen, ohne *Datenverarbeitungsspezialisten* einzuschalten.

Seibt

Anwendungsprogrammierer

Teilgebiet: Informatik in der Betriebswirtschaft *application programmer*

Synonym: Organisationsprogrammierer

Ein *Datenverarbeitungsspezialist*, der bei *Anwendern, Herstellern* oder *Software-Häusern* (→ *Software-Haus*) *Anwendungsprogramme* erstellt. Ergänzung: In der Praxis wird vielfach zwischen *Anwendungsprogrammierern* und *Systemprogrammierern* unterschieden. Beide Gruppen von *Programmierern* erhalten üblicherweise auch eine unterschiedliche Ausbildung.

Seibt

Anwendungsprogrammierung

Teilgebiet: Programmierung

Erstellung der benutzerinduzierten *Anwendungssoftware* und ihre *Wartung*.

Teilaufgaben sind:

- Programmwurf (→ *Programmwurf, strukturierter*),
- Formulierung in einer *Programmiersprache* und ggf. Übersetzung,
- *Programmtest*,
- *Dokumentation*.

Zur Unterstützung dieser Teilaufgaben stehen zahlreiche Methoden und Hilfsmittel zur Verfügung (siehe Bild S. 46):

Programmablaufplan [DIN 66001], *Programmaufbauplan* [vgl. Nassi, I; Shneiderman, B.: *Flowchart Techniques for Structured Programming*. In: SIGPLAN NOTICES, Vol. 8, Aug. 1973, S. 12-26] und *Entscheidungstabellentechnik* (*Entscheidungstabelle*) dienen dem *Programmwurf* und der *Dokumentation*. Ebenfalls auf die *Dokumentation*, in erster *Linie* aber auf den Entwurf und die *Niederschrift* des Programms, beziehen sich die Methoden der normierten [vgl. *DIN 66220*] oder *strukturierten Programmierung*. Gleiches gilt für *Programmgeneratoren* [vgl. *DIN 44300*, Nr. 69] auf der Basis dieser *Programmiermethoden* oder der *Entscheidungstabellentechnik*. Nur auf eine *Teilaufgabe* zugeschnitten sind *Testhilfen* (*Programme zur Fehlererkennung* und *-lokalisierung* (*Debugging*)). *Dialog-Programmiersysteme* erleichtern je nach *Leistungsfähigkeit* wenigstens die *Formulierung*, oft auch *Test* und *Dokumentation* von Programmen.

Mit diesen Methoden/Hilfsmitteln soll die *Anwendungsprogrammierung* im Sinne eines *„Software Engineering“* gestaltet werden, um einerseits die *Programme* vom Ersteller unabhängig werden zu lassen und andererseits die *Programmierer* zu entlasten.

Schmitz

Anwendungsprogrammierung als Organisationseinheit

Teilgebiet: Informatik in der Betriebswirtschaft *application programming department*

		Teilaufgaben			
		Entwurf	Niederschrift	Test	Dokumentation
Methoden/Hilfsmittel	manuell	Datenflußplan			x
		Programmaufbauplan	x		x
		Entscheidungstabellen	x		x
		Normierte Programmierung	x	x	x
		Strukturierte Programmierung	x	x	x
	maschinell	Programmgeneratoren		x	x
		Testhilfen			x
		Dokumentationssysteme			x
		Dialog-Programmiersysteme		x	x

Anwendbarkeit von Methoden und Hilfsmitteln auf die Teilaufgaben der Anwendungsprogrammierung.

Diejenige Gruppe oder Abteilung in der Organisationseinheit „Datenverarbeitung“, die für die Erstellung von *Anwendungsprogrammen* zuständig ist.

Ergänzung: Je nach Größe der Datenverarbeitung kann die Abteilung *Anwendungsprogrammierung* eine unterschiedlich große Anzahl von *Anwendungsprogrammierern* umfassen, als Abteilung verselbständigt und u.U. in Gruppen untergliedert sein. In großen betrieblichen DV-Abteilungen grenzt man z.B. die „Betriebswirtschaftliche Anwendungsprogrammierung“ von der „Technisch-mathematischen Anwendungsprogrammierung“ ab. Neben der Abteilung Anwendungsprogrammierung existiert häufig eine Abteilung „Systemprogrammierung“. *Seibt*

Anwendungsprotokoll

Teilgebiet: Informatik im Bankwesen

Protokoll der *Anwendungsschicht*. Nach ISO 7816-4 besteht das Protokoll von *Chipkarten* im Austausch von APDUs. *Martiny*

Anwendungsschicht → *Kommunikationssystem, offenes*

Anwendungssicherheitsmodul

Teilgebiet: Informatik im Bankwesen
Security Applicationer Module, SAM

Ein Gerät, welches Sicherheitsmechanismen für sonst unsichere Umgebungen bereitstellt. Es ist gegen Fälschungsversuche geschützt und speichert geheime Daten. *Martiny*

Anwendungssoftware

Teilgebiet: Programmierung
application software

Anwendungssoftware ist der Oberbegriff für alle Programme, die nicht Teil des *Betriebssystems (Systemsoftware)* sind. Die Programme der individuellen Anwendungssoftware lösen die aus den Zielen des *Anwenders* abgeleiteten, klar definierten *Datenverarbeitungsaufgaben*. Sie werden meist von diesem Anwender selbst erstellt und sind oft auf die konkrete Datenverarbeitungsanlage dieses Anwenders zugeschnitten.

Aufgrund ihrer individuellen Ausrichtung sind diese Programme für andere Anwender i. allg. nutzlos.

Programme der Standardanwendungssoftware (→ *Standardanwendungssoftware*) hingegen sind weniger spezialisiert und somit von mehreren Anwendern verwendbar:

Benutzerinduzierte Standardanwendungssoftware bezieht sich auf branchenspezifische Aufgabenstellungen im kommerziellen (z.B. Lohn-/Gehaltsabrechnung) oder mathematisch-technischen (z.B. Statik/ Dynamik) Bereich.

Systeminduzierte (→ *System'*) Standardanwendungssoftware stellt eine Erweiterung der Funktionen, die das Betriebssystem abdeckt, dar (z.B. *Generatoren, Monitore*); eine durchgehend saubere Abgrenzung zur Systemsoftware ist nicht möglich.

Als Ersteller von Standardanwendungssoftware kommen in erster Linie die *Hardware-Hersteller* und Softwarehäuser (→ *Software-Haus*) in Betracht.

Viele *Hersteller* bieten zusammen mit *Hardware* und Betriebssystem (= Systemsoftware) eine mehr oder weniger große Menge von mehrfach verwendbaren *Anwendungsprogrammen* an, die vom Anwender entweder kostenlos oder gegen

Zahlung einer Mietgebühr (*Unbundling*) benutzt werden kann. *Schmitz; Seibt*

Anwendungssystem

Teilgebiet: Programmierung

Siehe auch: System, soziotechnisches

Ein *soziotechnisches System*, das aus *Produkten* und der von ihrer Anwendung betroffenen Umgebung besteht. *Fritsche*

Anwendungssystem, rechnerintegriertes

Teilgebiet: Betrieb von Rechensystemen

computer-integrated application system

System, das einen Rechner für Steuerungs-, Datenverarbeitungs- oder Kontrollfunktionen als integrierten Bestandteil enthält. Der Rechner tritt nach außen hin nicht mehr direkt in Erscheinung, z.B. computergesteuerte (→ *computergestützt*) Fabriken, Flugüberwachungsgeräte und Hausgeräte. *Burkhardt*

Anwendungswissen → *Wissen, objektives¹*

Anzeige

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

display

Synonyme: Anzeigedarstellung; Datenanzeige

Visuelle *Datendarstellung* für die Kommunikation zwischen Mensch und Maschine. Die einfachste Form der gerätetechnischen (→ *Peripheriegerät*) Realisierung sind *binäre Darstellungselemente* (Signallampen), die nur zwischen den Schaltzuständen (→ *Schaltung, integrierte*) „Ein“ oder „Aus“ unterscheiden. Ziffernanzeigevorrichtungen stellen die nächst höhere Ebene der Datendarstellungsgeräte dar.

Für die *Mensch-Maschine-Kommunikation* in *computergestützten Informationssystemen* werden Anzeigegeräte für *alphanumerische* und *graphische* Datendarstellung eingesetzt. Wenn dabei die Möglichkeit der *Editierung* der *Anzeigedaten* im Dialogverkehr (→ *Dialogverarbeitung*) gegeben ist, spricht man auch von interaktiver (→ *Interaktion*) Anzeige, andernfalls von passiver Anzeige. *Encarnação; Lemke*

Anzeige, graphische

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

graphics; graphic display

Synonym: Datendarstellung, graphische

Anzeige von Daten, die sich mit *graphischen* Grundelementen wie Punkten, Geraden und Kurven darstellen lassen. Die graphische Anzeige kann dabei *statisch* oder *dynamisch* ausgeführt werden. Bei der statischen graphischen Anzeige

stehen die graphischen Daten für eine beliebige Zeit zur Ansicht zur Verfügung. Bei der dynamischen graphischen Anzeige werden Anteile der graphischen Daten in kleinen Zeitabschnitten kontinuierlich verändert und können somit als Bewegungsabläufe dargestellt werden.

Die gerätetechnische (→ *Peripheriegerät*) Realisierung der graphischen Anzeige konzentriert sich nach dem gegenwärtigen Stand der Technik (→ *Technik, Stand der*) auf punktschreibende und linienschreibende Verfahren. *Encarnação; Lemke*

Anzeige, interaktive

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

interactive display

Synonym: Datendarstellung, interaktive

Anzeige, welche zusätzlich zur visuellen *Datendarstellung* die Eingabe und das *Editieren* von *alphanumerischen* und/oder *graphischen* Daten im *Dialog* verfahren ermöglicht.

Die *interaktive* alphanumerische Anzeige erlaubt die Eingabe und das Editieren von Text (*Zeichenketten*) an einem Datensichtgerät.

Die interaktive graphische Anzeige erlaubt die Eingabe, *Ausgabe* und Manipulation (z.B. Editieren, Transformieren) von graphischen Daten an einem Bildschirmkonsolgerät, im Englischen oft auch als *interactive graphics* (→ *Anzeige, graphische*) bezeichnet.

Der Dialogverkehr (→ *Dialogverarbeitung*) wird durch spezielle *Eingabegeräte* ermöglicht. Gerätetechnisch werden die Eingabegeräte oft in fünf *logische* Klassen unterteilt:

Zeigestift (engl.: pick), alphanumerische Tastatur (engl.: keyboard), Wahltaste (engl.: button), Positionierer (→ *Positionierung*) (engl.: locator (→ *Lokalisierer*)) und Wertgeber (engl.: valuator). *Encarnação; Lemke*

Anzeigebereich → *Darstellungsfeld*

Anzeigedarstellung → *Anzeige*

Anzeigedaten

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

display data

Die Menge aller Daten, die für die *Anzeige* im Rechner abgespeichert sind. Abhängig von der gerätetechnischen Realisierung werden diese Daten direkt von einem Speicher oder nach weiterer *Verarbeitung* von spezieller *Hardware* (z.B. Transformation) für die *Darstellung(en)* verwendet. *Encarnação; Lemke*

Anzeigeelement

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

display element; display primitive

Synonym: Datendarstellungselement

Siehe auch: Darstellungselement

Gerätetechnisch eine *Hardware*-Funktion, die für die *Anzeigedarstellung* ausgeführt werden kann und einen sichtbaren Effekt am *Anzeigegerät* hervorruft, z.B. Punkt, Vektor, Zeichen und Kreisbogen. Abhängig von der *Hardware*-Funktion kann ein Anzeigeelement weiterhin mit einem oder mehreren Anzeigeelementen gekennzeichnet werden, z.B. Farbe, Leuchtintensität, Liniensstruktur, Liniendicke, Front, Zeichengröße, Zeichenraum und Identifizierungsschlüssel (→ *Identifikationsnummer*) (*Zeiger*).

Encarnação; Lemke

Anzeigegerät → *Anzeige*

Anzeigegruppe

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

display group

Eine Menge von Anzeigeelementen und/oder Anzeigesymbolen, die für die *Anzeigedarstellung* als eine Einheit zusammengefaßt sind. Diese Einheit kann mit weiteren Anzeigeelementen und/oder Anzeigegruppen zu übergeordneten Einheiten zusammengefaßt werden. Die *hierarchische* Anordnung der Elemente bleibt erhalten und ermöglicht eine *Editierung* der Elemente in der Gruppe.

In der *interaktiven Anzeige* kann die Anzeigegruppe als Einheit manipuliert (z.B. editiert und transformiert) werden. Dazu wird die Anzeigegruppe mit einem oder mehreren Anzeigeelement(en) gekennzeichnet, z.B. Identifizierungsschlüssel (*Zeiger*) und Darstellungstransformation.

Encarnação; Lemke

Anzeigehintergrund

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

background display; background image; static image

Der Teil einer *Anzeigedarstellung*, der vom Benutzer im allgemeinen nicht *interaktiv* verändert werden kann. Die *Formatanzeige* (→ *Formulareinblendung*) ist ein spezieller Fall des Anzeigehintergrunds und kann in der erstmaligen Erstellung des *Formats* u.U. *interaktiv* bearbeitet werden.

Encarnação; Lemke

Anzeigekonsolenoperator

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

display console operator

Eine Person, die ein *Anzeigegerät* benutzt und *interaktiv* mit Hilfe einer Anzeigekonsole (→ *Peripheriegerät*) mit dem Computer arbeitet.

Encarnação

Anzeigemenü → *Menü*

Anzeigeraum → *Darstellungsbereich*

Anzeigesymbol → *Anzeigegruppe*

AP, APL 2

Teilgebiet: Programmierung

Abkürzung von: A Programming Language

Eine *Dialogsprache*. APL wurde 1962 in dem gleichnamigen Buch von Iverson vorgestellt. Objekte, die mit dieser Sprache manipuliert werden können, werden als *Reihungen (Array)* von Komponenten verstanden, die nicht typgebunden sind. Die Sprache enthält eine Vielzahl flexibler *Operationen*, die gerade auf dieses Verständnis dieser Objekte ausgerichtet sind. APL fand Verbreitung durch das Ende 1966 fertiggestellte APL360-System zum interaktiven (→ *Interaktion*) Programmieren (auf IBM-Rechnern, später auch auf Rechnern anderer *Hersteller*), ein interpretierendes System (→ *Interpreterer*) mit sehr komfortabler Unterstützung des Benutzers. Nach einer größeren Zahl von Zwischenimplementierungen kam es 1982 zu einer Sprachentwicklung APL 2 mit allgemeiner strukturierten Objekten (Aufgabe der Beschränkung auf rechteckig angeordnete Reihungen).

APL wurde verschiedentlich auch zur Beschreibung von Rechnerstrukturen herangezogen.

Hoffmann, H.-J.

APDU

Teilgebiet: Informatik im Bankwesen

Abkürzung von: application protocol data unit
Dateneinheiten des Anwendungsprotokolls von Prozessorchipkarten. Man unterscheidet die Kommando-APDU vom *Terminal* an die *Chipkarte* und die Antwort-APDU von der *Chipkarte* an das *Terminal (ISO 7816-4)*. Die *Chipkarte* sendet ausschließlich nach einem *Kommando*.

Martiny

APLG

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

APLG ist eine Erweiterung von Iversons APL (→ *AP, APL 2*) für *graphische Ein- und Ausgabe*. APLG erhält man durch Erweiterung des kon-

ventionellen APL-Interpreters um eine Anzahl von *Standardfunktionen*, die der Durchführung der für die interaktiven (\rightarrow *Interaktion*) C.G.-Techniken (\rightarrow *Datenverarbeitung*, *graphische*) notwendigen *Operationen* wie Bilderzeugung, -manipulation und -verwaltung dienen.

Encarnação

APL-System \rightarrow AP, APL 2

Appel-Algorithmus

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

Durch den Algorithmus werden beliebige, durch ebene *Flächen* begrenzte Körper auf Visibilität getestet. Durch einen *Flächentest* wird zuerst die Zahl der zu testenden Kanten reduziert. Für alle noch verbleibenden Kanten wird die quantitative Unsichtbarkeit (Anzahl der Flächen, die ein *Punkt* verdeckt) untersucht. Da sich die quantitative Unsichtbarkeit einer Kante nur an den Schnittpunkten zwischen Kante und Konturlinie (*Linie* eines Körpers, die zu einer sichtbaren und einer unsichtbaren Fläche gehört) ändert, wird die Kante in Abschnitte zerlegt, die durch Anfangs- und Endpunkte der Kante sowie durch die Schnittpunkte der Kante mit Konturlinien definiert sind. Für jeden Kantenabschnitt (\rightarrow *Kante'*) wird die quantitative Unsichtbarkeit bestimmt. Ist sie gleich Null, so ist der Kantenabschnitt sichtbar, sonst unsichtbar.

Encarnação

application protocol data unit \rightarrow APDU

Application Specific Command \rightarrow ASC

Applikation \rightarrow Anwendung

applikative Programmierung \rightarrow Programmierung, applikative

Approximationsaufgabe

Teilgebiet: Quantitative Methoden

Bei einer Approximationsaufgabe wird für eine gegebene Funktion f eine approximative (angenäherte) *Darstellung* durch Funktionen eines bestimmten (i.d.R. einfacheren) Funktionstyps gesucht. Wir sprechen von diskreter Approximation, wenn der *Definitionsbereich* D der Funktion f nur diskrete Werte, und von kontinuierlicher Approximation, wenn D ein Zahlenkontinuum enthält (z.B. ein Intervall der reellen Zahlen). Häufig bildet die Menge der approximierenden Funktionen einen Vektorraum (z.B. bei der Approximation mit Polynomen); es handelt sich dann um eine lineare Approximationsaufgabe, andernfalls um eine nichtlineare (z.B. bei der Approximation mit rationalen Funktionen). Für die gestellte Aufgabe ist es wichtig, welches Maß für die Abwei-

chungen zwischen der Funktion f und der approximierenden Funktion P gewählt wird. Häufig werden die beiden folgenden Maße benutzt:

1. Die Tschebyscheff-Norm der Differenz $f - P$:

$$\|f - P\| = \sup_{x \in D} |f(x) - P(x)|$$

2. Die Norm der Fehlerquadrate:

$$\|f - P\|_2 = \sqrt{\sum_{x \in D} (f(x) - P(x))^2}$$

bei der diskreten und

$$\|f - P\|_2 = \sqrt{\int_D (f(x) - P(x))^2 dx}$$

(Im diskreten Fall müssen die Integrale durch Summen ersetzt werden). Bei Verwendung des ersten Maßes sprechen wir von Tschebyscheff-Approximation, bei Verwendung des zweiten von Gauß-Approximation oder auch Approximation im Sinne der kleinsten Fehlerquadrate.

Stahl

A Programming Language \rightarrow AP, APL 2

APSE

Teilgebiet: Programmierungstechnik

Ada Programming Support Environment, *Software-Entwicklungsumgebung* für die Entwicklung von Ada-Programmen.

Schneider

APT \rightarrow *Spezialsprache*

Arbeitsbereich

Teilgebiet: Grundlagen von Informationssystemen

working area; working space

Unter einem Arbeitsbereich kann man sich einen Zwischenspeicher oder ein *Terminal* vorstellen. Dieser Arbeitsbereich stellt eine Schnittstelle zwischen Datenteilsprache (\rightarrow *Datenmanipulationssprache*) und *Gastgebersprache* dar, da ein Benutzer die Gastgebersprache niemals direkt auf die Daten in der *Datenbank* anwendet, sondern diese zuerst mittels der Datenteilsprache in einen oder mehrere Arbeitsbereiche holt, um dann dort ganz bestimmte, nur mit einer Gastgebersprache mögliche *Operationen* (wie z.B. die arithmetischen Operationen) ausführen zu können.

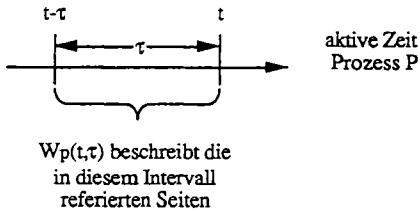
Schneider

Arbeitsgang \rightarrow *Arbeitsplan*

Arbeitsmenge

Teilgebiet: Betriebssysteme
working set

Die Arbeitsmenge $W_p(t, \tau)$ des Prozesses P zum Zeitpunkt t ist die Menge der Seiten, die in dem Zeitintervall $(t-\tau)$ referiert wurden. Das beschriebene Zeitintervall, das jeweils für die Zeit eines Prozesses im Zustand aktiv zu betrachten ist, wird in der nachfolgenden Abbildung veranschaulicht:



Für die zur Arbeitsmenge gehörigen Seiten entfällt mithin bei wiederholter Referenz das Nachladen aus dem Sekundärspeicher (\rightarrow *Hintergrundspeicher*). Man bezeichnet die Größe τ als die sogenannte Fenstergröße, die mit fortschreitender Zeit über die Zeitachse wandert. Zu beachten ist, daß die Arbeitsmenge zu einem bestimmten Zeitpunkt t jeweils für die τ vorangegangenen Zeiteinheiten verstanden wird. Richter, L.

Arbeitsmengen-Algorithmus

Teilgebiet: Betriebssysteme

Der Arbeitsmengen-Algorithmus gehört zur Klasse der dynamischen Seitenaustausch-Algorithmen, d.h. die einem Programm über die Zeit zugeordneten *Seitenrahmen* im *Hauptspeicher* sind in ihrer Anzahl nicht konstant. Bei gegebenem τ befinden sich zu jedem Zeitpunkt t genau diejenigen Seiten im Hauptspeicher, die zur *Arbeitsmenge* $W_p(t, \tau)$ gehören. Die exakte *Implementierung* des Arbeitsmengen-Algorithmus stößt in der Praxis auf Schwierigkeiten, da nämlich jedem Seitenrahmen im Hauptspeicher eine eigene *Uhr* zugeordnet sein müßte, die jeweils asynchron (\rightarrow *Asynchronität*) das *Betriebssystem* zur Verdrängung (Auslagerung aus dem Hauptspeicher) einer Seite veranlassen müßte, wenn die betreffende Seite während der letzten τ Zeiteinheiten nicht referiert wurde. Richter, L.

Arbeitsplan

Teilgebiet: Informatik in der Betriebswirtschaft
Arbeitsplan ist ein *Dokument* der betrieblichen Fertigungswirtschaft. Er legt produktbezogen die Art und die *Reihenfolge* der Bearbeitungsschritte zur Herstellung industriell gefertigter Güter fest.

Der Arbeitsplan untergliedert sich in Arbeitsgänge. Ein Arbeitsgang definiert den einzelnen Bearbeitungsschritt. Mit dem Arbeitsgang werden die vorgegebenen *Bearbeitungszeiten* ggf. aufgeteilt in *Rüst- und Ausführungszeiten*, der Bearbeitungsort, die Bearbeitungshilfsmittel (z.B. *Werkzeuge*) und die für die Bearbeitung erforderlichen Maschinen (\rightarrow *Automat*) festgelegt.

Im Rahmen *computergestützter* Fertigungsplanungsverfahren spielen die Arbeitspläne neben den *Stücklisten* eine dominante Rolle. Die Arbeitspläne sind dabei Grundlage für die Erstellung der Fertigungsbegleitpapiere, für die Durchlaufterminierung der Aufträge und für die Kapazitätsbelastungsrechnung (auch Kapazitätsbelegungsrechnung oder Maschinenbelegungsrechnung genannt). Bues

Arbeitsplanung, computergestützte

Teilgebiet: Informatik in der Fertigung

computer aided planning

Synonym: Arbeitsplanung, rechnergestützte

Teilfunktion der computergestützten Fertigung (CAM (\rightarrow *Assoziativspeicher*)) im Rahmen des CIM-Konzepts (\rightarrow *Computer Input by Microfiche/film*). CAP bedeutet die *computergestützte* Ableitung und Erstellung der Vorgaben für die Fertigung, i.e. Arbeitsplänen (*Arbeitsplan*) bei konventioneller Fertigung bzw. NC-Programmen bei computergesteuerten Produktionsanlagen (CNC-Maschine, *NC-Steuerung*), aus den *Stücklisten* und den geometrischen und technologischen Konstruktionsdaten, die in einem CAD oder PPS-System (\rightarrow *Produktionsplanungs- und -steuerungssystem*) abgelegt sind.

Hummeltenberg

Arbeitsplanung, rechnergestützte \rightarrow *Arbeitsplanung, computergestützte*

Arbeitsplanung, rechnerunterstützte

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

CAP; computer aided production planning

Rechnerunterstützte Informationsumsetzung während des Arbeitsplanungsprozesses. Ziel der Arbeitsplanung ist es, bei der Fertigung von Erzeugnissen ein Optimum aus Aufwand und Arbeitsergebnis zu erreichen. Ein rechnerunterstütztes System wird im wesentlichen zur Ausführung von Routinetätigkeiten und zur Bereitstellung von Informationen verwendet. Außerdem können verschiedene Lösungsalternativen untersucht und Optimierungsvorgänge durchgeführt werden. Dadurch werden Flexibilität, *Qualität* und *Transparenz* der Planung erhöht.

Krause, F.-L.; Spur, G.

Arbeitsplatz, graphischer

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung
workstation

Ein Arbeitsplatz, der aus einem *Ausgabegerät* zur Anzeige und Verarbeitung *graphischer* und *alphanumerischer* Daten und aus einem oder mehreren *Eingabegeräten* wie alphanumerischer Tastatur, Funktionstasten (→ *Auswähler*), *Steuerknüppel*, *Rollkugel*, *Lichtstift* (→ *Lichtgriffel*) usw. besteht (siehe auch *DIN 66233* „Bildschirmarbeitsplätze, Begriffe“). *Grieger*

Arbeitsplatzrechner → *Arbeitsplatz, rechnergestützt; Mikrocomputer; Heimcomputer*

Arbeitsplatz, rechnergestützter

Teilgebiet: Mikroelektronik
CAE station; work station

Synonym: Graphischer Arbeitsplatz für den IC-Entwurf

Speziell für den Entwurf von integrierten Schaltungen entwickelte man *Minicomputer*systeme mit meist farbigem *Bildschirm*. Durch die Verbindung von *Hard-* und *Software* und durch den Einsatz von *VLSI*-Schaltkreisen konnte der Preis drastisch reduziert werden. Die Entwicklung von kundeneigenen ICs (→ *Schaltkreis, integrierter*) wird dadurch erheblich gefördert. *Manck*

Arbeitsprogramm

Teilgebiet: Programmierung
work program

Der Programmteil, der die eigentliche *Verarbeitung* der Daten in einem größeren Programm bewerkstelligt, ohne Hilfs- und *Kontrollprogramme*. *Burkhardt*

Arbeitsrechner

Teilgebiet: Rechnernetze und verteilte Systeme
host; host computer

Synonyme: Dienstleistungsrechner; Verarbeitungsrechner; Wirtsrechner

Siehe auch: Server

Rechner innerhalb eines *Rechnernetzes*, dessen *Betriebsmittel* für die Bearbeitung von Aufträgen eingesetzt werden können. Die dafür zu erbringende DV-Leistung kann funktionell spezialisiert, d.h. auf die Bearbeitung bestimmter Arten von Aufträgen ausgelegt sein (*Nachrechner*). Als Arbeitsrechner werden oft auch nicht in Rechnernetze integrierte Rechner bezeichnet, die über ggf. intelligente *Datenstationen* (*Terminals*) zugänglich sind. *Wild*

Arbeitssicherheit → *Personalinformationssystem*

Arbeitsspeicher → *Hauptspeicher*

Arbeitsspeicherkapitel → *Datenteil*

Arbeitsvorbereitung

Teilgebiet: Betrieb von Rechensystemen
job preparation; production scheduling

Ausgehend von der Auffassung der Produktion in einem *Rechenzentrum* ist in Anlehnung zur industriellen Produktion auch in der *Informationsverarbeitung* vielerorts eine solche Funktion installiert. Produktionsreife *Anwendungsprogramme* werden der Arbeitsvorbereitung (AV) zur Betreuung im Produktionsbetrieb übergeben. Aufgaben der Arbeitsvorbereitung sind:

Einsatzvorbereitung, *Belegungsplanung*, Verwaltung der Arbeitsanweisungen für den Ablauf von Standardaufgaben, *Terminplanung* und Terminkontrolle, Datenarchivverwaltung und Vorbereitung von Routineprogrammläufen (→ *Programmlauf*).

Der Umfang der Aufgaben einer Arbeitsvorbereitung hängt von dem zu bearbeitenden Aufgabengebiet ab. Eine Unterstützung der Arbeitsvorbereitung erfolgt in zunehmendem Maße durch entsprechende *dialog*orientierte *Standardprogramme*. *Graef*

Arboreszenz → *Baum*

Architektur

Teilgebiet: Grundlegende Definitionen
architecture; system, architecture

Synonym: Systemarchitektur

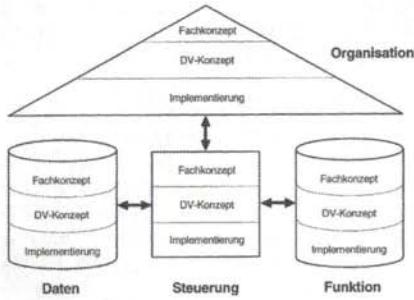
Das äußere Erscheinungsbild einer Anwendung, eines Programms, einer Maschine oder allgemein jeder Komponente eines *Datenverarbeitungssystems* wird beschrieben, indem für alle Eingabesituationen angegeben wird, „was“ als *Ausgabe* zu erwarten ist. Dieses „Was“ ist die Architektur, das funktionale Verhalten. Um eine übersichtliche Struktur zu erhalten, werden gegebenenfalls auch Teilergebnisse beschrieben. *Glathaar*

Architektur eines Informationssystems

Teilgebiet: Grundlagen von Informationssystemen

Die Architektur eines *Informationssystems* ist die Beschreibung seiner einzelnen Bausteine hinsichtlich ihrer Art, ihrer funktionalen Eigenschaften und ihres Zusammenwirkens. Neben dem Teil der *Architektur*, der die Komponenten und ihr Zusammenwirken definiert, ist auch das *Vorgehensmodell* zur Erstellung eines Informationssystems zu bestimmen. Eine Architektur bildet damit den Rahmen, in dem Informationssysteme entwickelt, optimiert und EDV-technisch realisiert werden können.

Ein Beispiel für eine Architektur ist die Architektur *integrierter Informationssysteme* (ARIS). Zur Reduzierung der *Komplexität* der Beschreibung von Informationssystemen werden in ARIS die Daten-, Funktions- und *Organisationssicht* gebildet. Die Darstellung der Verbindungen und der Zusammenhänge dieser Sichten wird in der *Steuerungssicht* vorgenommen.



Hirschmann; Scheer

Archivierung → *Datenhaltung, langfristige*

Archivsystem

Teilgebiet: Grundlagen von Informationssystemen

Integriertes System zur *Archivierung* von *Dokumenten*, meist als Ersatz von *Mikrofilm* eingesetzt. Besteht typischerweise aus einem Schwarzweiß-Scanner (200 oder 300 dpi), einem A4-Bildschirm, einem *Laserdrucker* und einem Steuerungsrechner (→ *Datenverarbeitungsanlage, digitale*) mit angeschlossener WORM-Einheit (→ *WRITE ONCE READ MANY*). Mentzel

arithmetische Operation → *Operation, arithmetische*

arithmetischer Ausdruck → *Ausdruck bei Programmiersprachen*

arithmetisches Mittel → *Mittelwert*

arithmetische Verzweigung → *Verzweigungsanweisung (FORTRAN)*

arithmetisch-logische Einheit → *Einheit, arithmetisch-logische*

Arithmetisierung → *Gödelisierung*

ARM → *Spontanbetrieb*

ART → *Automated Reasoning Tool*

Artfestlegung → *Typisierung*

Artificial intelligence → *Intelligenz, künstliche*

Artificial Reality

Teilgebiete: Wissensverarbeitung und Expertensysteme; Computer Graphics, CAD und Bildverarbeitung

Synonym: virtual reality

Artificial Reality bezeichnet die Bemühungen, dem Computerbenutzer durch Ausschöpfen aller technischen Möglichkeiten der Ausgabe- und Eingabegeräte visuelle, auditive und taktile Eindrücke so zu vermitteln, daß ein ganzheitlicher Eindruck einer künstlichen Welt entsteht, in der der Benutzer sich bewegen und mit Objekten interagieren kann. Dabei werden insbesondere stereoskopische und stereophonische Ausgaben verwendet, die meist über einen *Datenhelm* vermittelt werden. Neben Kopfhörern und einem Mikrofon sind in diesem Helm zwei LCD-Bildschirme unmittelbar vor den Augen des Trägers angebracht. Oft wird auch ein Meßgerät für die Kopfbewegungen integriert. Die *Interaktion* des Benutzers mit der künstlichen Welt erfolgt durch spezielle Eingabegeräte wie *Datenhandschuh* (data glove) und *Datenanzug* (data suit) zur Übertragung von Gestik oder *Spracheingabe* zur Übertragung von Befehlen. Dazu werden Meßfühler so im Handschuh oder im Anzug angebracht, daß die Dehnungen über den Körpergelenken gemessen werden, um daraus die absoluten *Koordinaten* der Körperteile zu berechnen. Artificial Reality Systeme werden in Simulatoren (→ *Simulation*) eingesetzt, in denen die Handhabung und Steuerung von komplexen technischen Systemen insbesondere in gefährlichen Situationen trainiert werden soll, z.B. das Manövrieren von Raumfahrzeugen und Flugzeugen, sowie das Operieren mit Endoskopie-Laserskalpellen. Populär geworden sind Artificial Reality Systeme durch Spielgeräte. Gorny

Artware → *Computerkunst*

ASC

Teilgebiet: Informatik im Bankwesen

Abkürzung von: Application Specific Command

Kommandos, die für eine Anwendung benötigt werden, aber nicht vom *Betriebssystem* standard bereitgestellt werden. EN 726-3 beschreibt Mechanismen für ein gesichertes Einbringen von ASCs. Martiny

ASCII

Teilgebiet: Programmierungstechnik

Synonym: USASCII

Abkürzung von: American Standard Code for Information Interchange

Die nationale Variante des *ISO-7-Bit-Codes* in den USA. Der ASCII-Code hat mit der Verbrei-

B7 B6 B5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
B4 B3 B2 B1	Steuerung			Ziffern Symbole		Großbuchstaben		Kleinbuchstaben
0	NUL	DLE	SP	0	@	P	'	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
10	LF	SUB	*	10	J	Z	j	z
11	VT	ESC	+	11	K	[k	[
12	FF	FS	,	12	L	\	l	l
13	CR	GS	-	13	M]	m]
14	SO	RS	:	14	N	^	n	^
15	SI	US	/	15	O	_	o	_

octal	25	ASCII-Zeichen
hex	15	

Der ASCII-Code ist eine Weiterentwicklung der amerikanischen Computertechnologie, die weltweit Bedeutung erlangt. Die Codes sind stets durch ein 8. Bit, das Paritätsbit (→ Parity-bit), ergänzt.
Burkhardt; Matejka

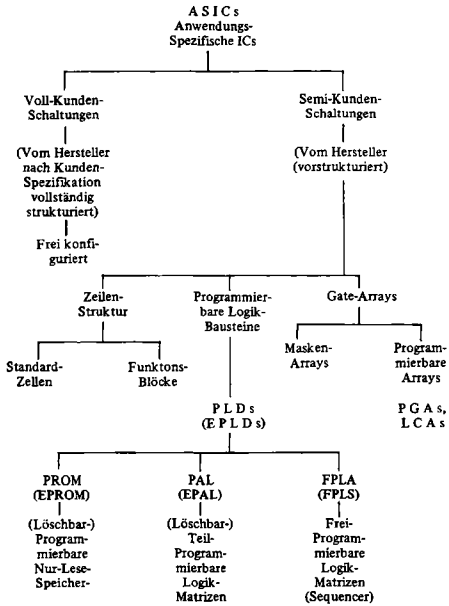
ASCII-Code → USASCII-Code

ASIC

Teilgebiete: Mikroelektronik; Rechner-technologie; Schaltwerke und Schaltnetze
application specified integrated circuit
Abkürzung von: Integrierte Schaltungen, anwendungsspezifische

ASICs entwickelten sich in Abstimmung zwischen den funktionell sowie technologisch inter-

essierenden Entwicklungszielen und den ökonomischen Notwendigkeiten. Sie umfassen neben den Voll- eine verzweigte Familie von Semi-Kundenentwürfen. Hierfür gilt u.a.: möglichst vielseitig verwendbare Typen zugunsten hoher Stückzahlen. Die Familie setzt sich aus Entwurfsstilen mit unterschiedlichen Merkmalen (→ *Attribut*) zusammen. Darunter ist z.B. das Festlegen des ASIC-Verhaltens durch *Hersteller* oder *Anwender* bzw. Designhaus unterschiedlich aufgeteilt. Die Abb. S. 54 zeigt eine hierarchische Übersicht; darin sind folgende, bereits getrennt definierte Begriffe zusammengefaßt und ergänzt: *Standardzellentwurf*, *Gate-Array*,



PROM, PAL, PLA (programmierbare logische Anordnung), Schaltungen (→ Schaltung, integrierte), integrierter Kundenentwurf.

Der logische und physikalische Entwurf von ASICs sowie die Testunterstützung (→ Testen) haben sich durch zahlreiche CAD/CAE-Werkzeuge gewandelt. Diese können z.T. große materielle und personalqualitative Investitionen erfordern. Entwurfsoptimierungen nach Signallaufzeit oder nach Siliziumfläche sind Beispiele von Softwareleistungen. Mit Simulationen sind sie bereits vor der Programmfestlegung in Anwendernähe (Programmiergeräte) testbar: PLDs, d.h. Programmable Logic Devices.

Eine Übersicht der ASIC-Entwurfsbereiche und -ebenen im Y-Modell, ein sog. Gajski-Diagramm, läßt die nicht ohne weiteres kompatiblen Einstiegsmöglichkeiten in den Entwurf erkennen. Diese können durch CAD/CAE-Systeme z.T. ausgefüllt werden. Übergänge zwischen den Entwurstilen erfordern besondere Aufwendungen.

Für die Wahl einer ASIC-Art sind u.a. folgende Parameter maßgebend: Größenordnung und Komplexität des Bausteins, Stückzahlabschätzungen, verfügbare Zeit und Kosten der Entwicklung, verfügbares CAD/CAE-System. Außer dem Funktionsumfang einer Anwendung beeinflußt die kleinste programmierbare Grundeinheit eines ASICs die Baueingröße (→ Baueinheit) und -komplexität.

Anwendungsbeispiele für ASICs sind Bausteine

mit unterschiedlichen Komplexitäten für: Rechnerarchitekturen, Peripheriegeräte, Telekommunikation, Prozeßautomatisierungen, Sicherheits- und Schutzsysteme, Fahrzeug- und Flugzeugautomatisierungen, Haushaltselektronik, Unterhaltungselektronik Meintzen

Assembler → *Assemblierer*

Assemblerbefehl

Teilgebiet: Programmierung

Programmieranweisung (Befehl), die von der Hardware direkt in einem Maschinenbefehl ausgeführt werden kann, die aber in symbolischer Form (Code, mnemonischer) angegeben wird. Wegen der 1:1 -Entsprechung zwischen Assembler- und Maschinenbefehl ist die *Assemblersprache* (die Summe der für eine Maschine definierten Assemblerbefehle (→ *Assemblierer*)) für jede CPU-Architektur anders. Assemblerbefehle erlauben unbedingte und, von einer Vorbedingung abhängige, bedingte Sprünge (→ *Sprungbefehl*) an eine symbolisch (mit einem Namen (→ *Bezeichner*)) angegebene Programmadresse; ausserdem können in einer Assemblersprache i.a. *Makros* definiert werden. Nowak

Assemblersprache

Teilgebiete: Programmierung; Übersetzerbau *assembly language*
Synonym: Symbolcode

Kunstsprache vom algorithmischen Typus (*algorithmische Programmiersprache*), die die Formulierung eines Programms in Konstrukten vorsieht, die strukturisomorph oder zumindest strukturäquivalent zu den Befehlen einer bestimmten Rechenanlage sind und, ohne Zwang zur Einhaltung von Artkonsistenz (*Typisierung*), nur Speicherplätze (und *Register*) der Anlage anzusprechen erlauben. Im Gegensatz zur *Maschinsprache* der Anlage sieht die *Assemblersprache* jedenfalls die Verwendung *symbolischer Adressen* vor und muß daher auch deklarative Konstrukte zur Symboldefinition bereitstellen. Häufig kennt eine *Assemblersprache* auch Konstrukte, die den Ablauf des *Assembliervorgangs* steuern (also keinem *Maschinenbefehl* zuzuordnen sind). Unter Umständen erlaubt sie auch eine gewisse *Abstraktion* von unmittelbaren Maschineneigenschaften (→ *Maschinenabhängigkeit*), insbesondere zur Inanspruchnahme der Dienstleistungen eines *Betriebssystems*, in Form von *Makrobefehlen*. Der Gebrauch einer *Assemblersprache* erfordert das Vorhandensein eines *Assemblerprogramms* (→ *Assemblierer*).

Hoffmann, H.-J.; Lehmann

Assemblierer

Teilgebiet: Programmierung

assembly program

Synonyme: Assembler; Assemblierprogramm

Ein *Übersetzer*, der in einer *Assemblersprache* formulierte Programme in bedeutungstreue Programme in der *Maschinensprache* der für die Ausführung der Programme vorgesehenen Rechenanlage umwandelt.

Der Vorgang der Übersetzung wird als Assemblierung (bzw. Assembliervorgang) bezeichnet. *Hoffmann, H.-J.; Lehmann*

Assemblierprogramm → *Assemblierer*

Assemblierung → *Assemblierer*

Assembliervorgang → *Assemblierer*

ASSIGN → *Verzweigungsanweisung (FORT-RAN)*

Assistent, mathematisch-technischer → *Org-ware*

Assoziation

Teilgebiet: Grundlagen von Informationssystemen

Eine Menge von zwei oder mehr Objekt-Rollen-Paaren. Jedes der beteiligten Objekte spielt im Rahmen der Assoziation eine bestimmte, festgelegte Rolle. Assoziationen sind die semantisch relevanten Bestandteile eines *Informationsbereichs*, z.B. die Assoziation, die zwischen einer *Entität* – repräsentiert durch einen Satz – und seinen Attributwerten bestehen. Oder: die Beziehungen zwischen Sätzen (→ *Satz¹*) in einer *Datenbank* (siehe SET). *Schneider*

assoziativ → *Assoziativspeicher*

Assoziativprozessor

Teilgebiet: Rechnerarchitektur

associative processor

Synonym: Assoziativrechner

Als Assoziativprozessoren oder Assoziativrechner werden im allgemeinen *Prozessoren* bezeichnet, die die folgenden Eigenschaften haben:

1. Arbeitsspeicher (→ *Hauptspeicher*) ist ein *Assoziativspeicher*, und die gespeicherten Informationen (evtl. auch die zugehörigen Programme) sind durch Angabe von Gesamt- oder Teilinhalten wiederauffindbar bzw. manipulierbar.
2. Zur Verfügung stehen Suchoperationen (→ *Suchverfahren¹*) auf der Basis *logischer* und *arithmetischer* Vergleiche, die typischerweise auf einer Vielzahl von *Operanden* gleichzeitig ausgeführt werden.

Assoziativprozessoren gehören damit zur Klasse der SIMD Single Instruction Stream Multiple Data Stream-Prozessoren. Je nach Aufbau der *Hardware*, Einsatzgebiet und *Programmlaufsteuerung* unterscheidet man zwischen voll parallelen (→ *parallel²*), *bitseriellen*, wortseriellen und block-orientierten (→ *Block²*) Assoziativprozessoren. *Ameling; Giloi*

Assoziativrechner → *Assoziativprozessor*

Assoziativspeicher

Teilgebiet: Rechnerorganisation

associative memory; content addressable memory

Synonym: Speicher, inhaltsadressierbarer

Abkürzung: CAM

Ein Speicher, dessen *Speicherzellen* durch Angabe ihres Inhalts oder eines Teils davon adressiert werden, bei dem also der Inhalt genau der Speicherzellen ausgegeben wird, in denen eine vorgegebene Schlüsselinformation vorkommt.

Ein Assoziativspeicher umfaßt üblicherweise neben der eigentlichen Speichereinheit (→ *Speichermodul*) ein *Suchregister*, ein *Maskenregister* und eine sog. *Wortrandlogik*. Bei der *Adressierung* wird der Inhalt des Suchregisters in den durch das Maskenregister ausgewählten *Bitpositionen* mit den korrespondierenden Bits sämtlicher Speicherwörter (→ *Speicherstelle*) zeitparallel (→ *nebenläufig*) verglichen. Dazu ist jedem *Speicherelement* eine Vergleichslogik zugeordnet; die Ausgänge der Vergleichslogiken werden im Speicher wortweise ausgewertet. Bei *Übereinstimmung* des Suchworts (→ *Suchen*) mit einem oder mehreren *maskierten* Zelleninhalten wird dies über sog. *match lines*, die den Zellen zugeordnet sind, angezeigt und in der *Wortrandlogik* gespeichert. Die *Wortrandlogik* besitzt dazu pro Speicherzelle ein *Speicherelement*. Bei Mehrfachanzeigen kann die *Reihenfolge* der Auswertung nach *Prioritäten* vorgenommen werden. Dies kann durch die *Wortrandlogik* unterstützt werden, wenn diese ein *Prioritätenschaltznetz* enthält. Beim Einsatz von Assoziativspeichern als *Caches* (→ *Pufferspeicher*) werden die Speicherwörter in je einen *Daten-* und einen *Adreßteil* (*tag*, Etikett) unterteilt. Außerdem erfolgt die Speicherbelegung so, daß dieselbe Adresse nicht mehrfach gespeichert ist. Das *Maskenregister* und die *Wortrandlogik* entfallen hierbei. *Flik; Giloi*

astabil → *Kippschaltung*

astabile Kippschaltung → *Kippschaltung*

astabiles Flipflop → *Multivibrator*

asymmetrisches Multiprozessorsystem

→ *Multiprozessorsystem*

asynchroner Kommunikations-Schnittstellen-Baustein → *Kommunikations-Schnittstellen-Baustein, asynchroner*

asynchroner Zähler → *Zähler, asynchroner*

asynchrones Schaltwerk → *Schaltwerk, asynchrones*

asynchrone Übertragung → *Übertragung, asynchrone*

Asynchronität

Teilgebiet: Rechner-technologie
asynchronism

Die Eigenschaft eines Netzwerks (→ *Rechner-netz*), ohne *Takt* zu arbeiten oder von mehreren zusammengeschalteten Netzwerken mit je einem eigenen Takt. *Burkhardt*

Asynchronous Transfer Mode

Teilgebiet: Datenübertragung
Synonym: ATD
Abkürzung: ATM

Mit ATM wird eine Übertragungstechnik bezeichnet, die sehr viele digitale Datenströme quasiseparabel auf einer einzelnen seriellen Leitung überträgt. ATM ist das für das Breitband-ISDN (→ *B-ISDN*) geplante Übermittlungsverfahren. Die ATM-Technik ermöglicht für digitale Sprach- (Audio) und Bilddaten (Videodaten) eine für den Nutzer kontinuierliche Online-Übertragung. Die *Teilnehmer* kommunizieren über *virtuelle* Verbindungen, die über eine spezielle Signalisierungsverbindung aufgebaut, abgebaut und verwaltet werden. An der Teilnehmer-Netz-Schnittstelle werden alle Daten in kontinuierlich übertragene Zellen (→ *Speicherzelle*) gesendet und *empfangen*. Die Länge aller Zellen ist gleich, sie beträgt 53 Bytes. Dabei sind 5 Bytes Header und 48 Bytes Daten vorgesehen. Jede Zelle kann mit Daten einer beliebigen Verbindung gefüllt sein, zur Zuordnung enthält sie eine Verbindungskennung. Die Anzahl der Verbindungen an einer Teilnehmer-Netz-Schnittstelle ist durch die Gesamtbitrate der Schnittstelle begrenzt.

Aufeinanderfolgende Zeilen eines *Nachrichtenstroms* werden in äquidistanten Abständen übertragen; dadurch wird die kontinuierliche Online-Übertragung ermöglicht.

Beim *Verbindungsaufbau* werden die Bitrate der Verbindung, der Weg für alle zu übertragenden Daten und eine Kennung für jeden Übertragungsabschnitt festgelegt. Das Unterschreiten der Bitrate ist zulässig, eine Überschreitung in einem

bestimmten Rahmen ebenfalls. Auf die Reservierung von Übermittlungsressourcen im Netz wird weitgehend verzichtet. Die *Verzögerung* der Daten kann daher während einer Verbindung schwanken, Zellenverlust und -fehlzustellung sind möglich.

ATM arbeitet in der Regel in Glasfasernetzen mit einer Datenübertragungsrate (→ *Datenübertragungsrate*¹⁾) von insgesamt 155 *Mbit/s*, die Übertragungsrate der einzelnen Nutzer kann variiert werden, indem mehrere Zellen pro Übertragungszyklus für einen Nutzerstrom zusammengefaßt genutzt werden. *Nilsson; Popescu-Zeletin*

Asynchronverfahren

Teilgebiet: Datenübertragung
asynchronous operation

Synonym: Start-Stop-Übertragung

Eine Übertragungsart, bei der die *Binärzeichen* einer Übertragungszeichenfolge in einem festen Zeitraster liegen und bei der für diese Übertragungszeichenfolge zwischen den *Datenstationen* *Synchronismus* besteht.

Die Binärzeichen verschiedener Übertragungszeichenfolgen müssen nicht im gleichen Zeitraster liegen.

Die *Bitsynchronisation* wird bei Empfang des *Startbits* hergestellt und für die Dauer des zugehörigen Zeichens (Übertragungszeichenfolge) aufrechterhalten. *Matejka*

Asynchronzähler → *Zähler als Schaltwerk*

ATD → *Asynchronous Transfer Mode*

ATM → *Asynchronous Transfer Mode*

ATN → *Augmented Transition Network*

atomar

Teilgebiete: Betriebssysteme; Rechnerorganisation
atomic

Eine Eigenschaft von Aktionen des *Zugriffs* auf gespeicherten Daten in dem Sinne, daß derartige Aktionen *Operationen* auslösen, die unteilbar sind. Diese als unteilbar anzusehenden Operationen können (in einer anderen Betrachtungsebene) sehr wohl aus der Ausführung von entsprechenden Befehlen, Anweisungen, *Kommandos* oder von daraus gebildeten Folgen zusammengesetzt sein. Mehrere gleichzeitig ausgelöste atomare Zugriffe auf dasselbe *Datenobjekt* schließen sich gegenseitig aus. Die *Reihenfolge* der Ausführung von gleichzeitig ausgelösten Operationen dieser Art ist zufällig. Bei Datenobjekten, die zusammengesetzten *Datentypen* entsprechen, kann die gesamte Zugriffsaktion unter Ausnut-

zung der Zugriffe auf einzelnen Komponenten in *Phasen* zerlegt werden, die ihrerseits unteilbar sind. Dabei werden zumeist Konsistenzbedingungen wirksam, die dem Datenobjekt zugeordnet sind, auf das zugegriffen werden soll (*Quelle*: Informatik-Spektrum Bd. 7, 1984, S. 247/248). *Fleischhauer; Guse; Rouette*

atomare Formel → *Formel, atomare*

Atomformel → *Hornlogik*

ATR → *Answer to reset*

Attribut

Teilgebiet: Grundlagen von Informationssystemen

attribute

Synonym: Merkmal²

Eine mit einem Namen (→ *Bezeichner*) versehene Eigenschaft einer *Entität*. Attribute sind sowohl informationsseitig betrachteten Entitäten (in der realen Welt) als auch datenseitig betrachteten Entitäten zugeordnet. Meistens kommt ihre Bedeutung in ihren Namen zum Ausdruck (nach ISO/IEC 2382-17).

Beispielsweise informieren die Attribute NAME, VORNAME, STRASSE, HAUSNUMMER und ORT über den Wohnsitz einer Person (in der realen Welt) sowie über die Bedeutung der Ausprägungen entsprechend benannter *Datenfelder* in *Datensätzen* einer Adreßdatei, die einzelnen Personen zugeordnet sind.

Mathematisch läßt sich ein Attribut verstehen als eine Beziehung (*Relation*) über den Grundmengen Objektklasse (→ *Klasse*²) und Attributwerte (aus einem vereinbarten oder als vereinbart geltenden *Wertebereich*).

Im relationalen *Datenmodell* wird ein Attribut durch die Spaltenüberschrift einer *Tabelle* repräsentiert.

Anmerkung: Die Benennung Objektklasse entspricht anglo-amerikanischem Sprachgebrauch und verwendet das Wort Klasse, obwohl dessen Begriffskonzept und Definition nicht mit dem mathematischen Klassenbegriff verträglich sind. *Barkow; Fleischhauer; Hesse et al.*

Attributgrammatik

Teilgebiet: Automatentheorie und formale Sprachen

attribute grammar

Synonym: Attributierte Grammatik, Affix-Grammatik

Eine Attributgrammatik über einer *kontextfreien*, formalen *Grammatik* $G = (N, T, R, S)$ besteht aus

- einer Zuordnung von zwei disjunkten Mengen, nämlich $Inh(X)$, der Menge der ererb-

ten („inherited“) *Attribute*, und $Syn(X)$, der Menge der abgeleiteten („synthesized“) *Attribute* zu jedem Symbol $X \in (N \cup T)$;

- der Festlegung eines Wertebereichs D_a für jedes Attribut a ;
- der Angabe *semantischer Regeln*, die für jede *Produktion* $(X \rightarrow v) \in R$ und jedes Vorkommen (Y, i) eines Symbols Y an i -ter Position in v in Form von Funktionen festlegen,
 - wie sich die abgeleiteten Attribute von X aus den Attributen aller (Y, i) berechnen; und
 - wie sich jedes ererbte Attribut eines Vorkommens (Y, i) aus den Attributen von X und denen der sonstigen Symbolvorkommen in v berechnen.

Attributgrammatiken zeichnen sich gegenüber kontextfreien Grammatiken durch eine größere Beschreibungsmächtigkeit aus und sind das bei der Generierung von *Übersetzern* gebräuchlichste Mittel zur Behandlung der statischen semantischen Analyse (→ *Erschließung, inhaltliche*). *Partsch*

Attributierte Grammatik, Affix-Grammatik

→ *Attributgrammatik*

Attributklasse

Teilgebiet: Grundlagen von Informationssystemen

attribute class

Synonym: Merkmalsklasse

In einem gegebenen Betrachtungszusammenhang die Menge aller möglichen *Attributwerte* (Merkmalswerte) ein und derselben Eigenschaft der tatsächlich auftretenden *Entitäten* einer *Entitätsklasse*.

Beispielsweise bilden in einer Personaliendatei über alle Betriebsangehörigen eines Unternehmens alle Kalenderdaten ab 1900 als mögliche Attributwerte des *Attributs* Geburtsdatum eine Attributklasse. Das begriffliche Umfeld stellt sich so dar: Überlegungen zur Gestaltung von *Informationssystemen* gehen häufig von einem durch die Betrachtungsweise abgegrenzten und hervorgehobenen „Weltausschnitt“ (univers of discourse) oder Betrachtungszusammenhang aus. Dabei sollen insbesondere solche Entitäten mit ihren Attributen (Merkmalen) und *Assoziationen* behandelt werden, die bei eben dieser Betrachtungsweise, also in diesem Betrachtungszusammenhang, von Interesse sind. In der Gesamtheit der interessierenden Entitäten lassen sich Entitätsklassen und in der Gesamtheit der den Entitäten zugeordneten Attribute lassen sich Attributklassen finden, die bei einer beabsichtigten

Attribut(-typ)

Strukturierung zugehöriger Daten von Bedeutung sind (nach ISO/IEC 2382-17).

Anmerkung: Die Benennung entspricht anglo-amerikanischem Sprachgebrauch und verwendet das Wort Klasse, obwohl Begriffskonzept und Definition nicht mit dem mathematischen Klassenbegriff verträglich sind.

Fleischhauer; Guse; Rouette

Attribut(-typ)

Teilgebiet: Grundlagen von Informationssystemen

attribute (type)

Synonym: Attributtyp, Eigenschaftstyp, Eigenschaft

Ein *Attribut* ist eine Zuordnungsvorschrift (\rightarrow *Zuordnungsproblem*), die jede *Entität* einer *Entitätsmenge* zu jedem Zeitpunkt jeweils höchstens ein Element aus einem *Wertebereich* zuordnet.

Bemerkung: Attribute dienen zur Beschreibung von Eigenschaften von Gegenständen mit Hilfe von Werten aus vorgegebenen Wertebereichen. Formal kann man ein Attribut definieren als Funktion (mit Entitätsmenge *EM*, Zeit *T* und Wertebereich *W*):

$f: EM \times T \rightarrow W$.

Bemerkung: Im *relationalen Datenmodell* wird ein Attribut durch die Spaltenüberschrift einer *Tabelle* repräsentiert.

Beispiel: Durch das Attribut Geburtstag ist jedem Kunden ein Element aus dem Wertebereich Datum zugeordnet. *Barkow; Hesse; Kittlaus et al.*

Attributtyp, Eigenschaftstyp, Eigenschaft

\rightarrow *Attribut(-typ)*

Attributwert

Teilgebiet: Grundlagen von Informationssystemen

attribute value

Ein konkretes Vorkommen (tatsächliches Auftreten) eines *Attributs*. Dies gilt sowohl informativ (also in der realen Welt) als auch datenseitig. In aller Regel ist nicht die Gesamtmenge aller denkbaren Attributwerte von Interesse, sondern nur eine von der jeweils in Betracht gezogenen Objektklasse bestimmte und abgegrenzte Teilmenge, die als *Wertebereich* ggf. zu vereinbaren ist (nach ISO/IEC 2382-17).

Beispiel: Im *Katalog* der RAL-Farben wird die Farbe „ziegelrot“ genannt; Attribut ist „Farbe“; Attributwert ist „ziegelrot“.

Sofern jedem Element der Objektmenge (Objektklasse) genau ein Attributwert zugeordnet ist, werden so Paare aus einem Element der Objektklasse und einem Attributwert gebildet, die ins-

gesamt eine Paarmenge ausmachen, welche (mathematisch verstanden) durch das Attribut repräsentiert wird.

Im *relationalen Datenmodell* werden Attributwerte durch *Tabelleinträge* repräsentiert.

Anmerkung: Die Benennung Objektklasse entspricht anglo-amerikanischem Sprachgebrauch und verwendet das Wort Klasse, obwohl dessen Begriffskonzept und Definition nicht mit dem mathematischen Klassenbegriff verträglich sind. *Barkow; Fleischhauer; Hesse et al.*

Audiovision

Teilgebiet: Grundlegende Definitionen

Wiedergabe bzw. Aufzeichnung von Bild- und Tonprogrammen mit Hilfe von *Datenträgern* wie Film, *Kassette*, *Band*, *Platte* (\rightarrow *Magnetplatten-speicher*). *Fuchs*

Audio-visuelles Medium

Teilgebiet: Grundlegende Definitionen

Abkürzung: AV-Medium

Audio-visuelle Medien sind u.a. Videorecorder, Videokamera, Videospiele, Großprojektoren, *Bildplatte* oder Abtastgeräte und Compact Disk (CD). *Fuchs*

Audit \rightarrow Qualitätsaudit

Aufbauorganisation \rightarrow *Informationsorganisation; Organisation*

Auffangflipflop \rightarrow *Taktsteuerungsarten eines Flipflops*

Aufforderungsbetrieb

Teilgebiet: Datenübertragung
normal response mode; NRM

Bei *bitorientierten* Steuerungsverfahren eine Betriebsart (\rightarrow *Rechensystem, Nutzungsformen eines*), in der die Folgesteuerung nur nach Aufforderung durch die Leitsteuerung DÜ-Blöcke (\rightarrow *Frame in der Datenübertragung*) *senden* kann. Die Aufforderung wird mit *Steuerdaten* gegeben. *Eckert*

Auffrischen

Teilgebiete: Mikroelektronik; Rechnerorganisation

Periodisches *Ausgleichen* der Ladungsverluste der in *dynamischen Speichern* verwendeten Kondensatoren zur Regeneration der Information.

Burkhardt

Auffrischrate \rightarrow *Bildwiederholungsrate*

Aufgabenangemessenheit

Teilgebiet: Programmierung
task adequacy

Ein *Dialog* ist aufgabenangemessen, wenn er die Erledigung der Arbeitsaufgabe des Benutzers unterstützt, ohne ihn durch Eigenschaften des Dialogsystems unnötig zu belasten. Tätigkeiten, die sich aus der technischen Eigenart des Dialogsystems ergeben, sollen im allgemeinen durch das System selbst ausgeführt werden. *Bullinger*

Aufgaben betrieblicher Rechenzentren → *Datenverarbeitungszentren, Aufgaben betrieblicher*

Aufgaben-Definition

Teilgebiet: Programmierung

Gesamtheit aller Tätigkeiten, die zur Festlegung der Anforderungen, Festlegung der Eigenschaften, Klärung der Realisierbarkeit und Festlegung des globalen Lösungsweges eines *Produktes* oder Teilproduktes erforderlich und keine Management-Tätigkeiten sind.

Anmerkung: Der Aufgaben-Definition können weitere Tätigkeiten wie z.B. die Durchführung von Markt- und Problemanalyse, Organisationsuntersuchung oder Studien vorausgehen.

Fritsche

Aufgabendefinition, fachlicher/funktionaler/sachlogischer Entwurf, Fachspezifika

→ *Fachkonzept*

Aufgabengenerator

Teilgebiet: Computergestützter Unterricht und Pädagogik
problem generator

In Übungs- und *Testprogrammen* des Computerunterstützten Unterrichts werden bestimmte Aufgabentypen verwendet; durch zufällige Variation ihrer Größen können für eine große Zahl von Adressaten individuelle Aufgaben erzeugt werden.

Ein Aufgabengenerator variiert dabei die Eingabegrößen mit Hilfe eines (Pseudo-) Zufallsgenerators (→ *Zufallszahlengenerator*) und generiert die individualisierten Aufgaben und deren Lösungen.

Gunzenhäuser

Aufgabenstrom

Teilgebiet: Betriebssysteme
job stream

Synonym: Auftragsstrom

Die Folge von Aufträgen, die ein gegebenes System innerhalb einer bestimmten Zeitperiode abarbeiten soll. Der Aufgabenstrom beinhaltet je nach Systemkonfiguration (→ *Konfiguration eines Datenverarbeitungssystems*) Anweisungen der *Kommandosprache*, Programme und Daten. Durch Kommando-Prozeduren kann der tatsächliche Inhalt des Aufgabenstroms ergänzt oder modifiziert werden. Physikalisch gesehen kann

der Aufgabenstrom von einem oder mehreren *Eingabegeräten* aus dem System zugeführt werden; er kann aber auch im Dialogbetrieb im Verlaufe eines *Gesprächs* spezifiziert worden sein.

Endres

aufladen

Teilgebiet: Informatik im Bankwesen

Tätigkeit, die eine elektronische Börse auffüllt. Der Vorgang wird durch ein *Aufladeprotokoll* gesichert.

Martiny

Aufladeprotokoll

Teilgebiet: Informatik im Bankwesen

Ablaufvorschrift (→ *Ablauf eines Programms*), die beim Aufladen einer elektronischen Börse eingehalten werden muß. Ein Aufladeprotokoll beinhaltet im allgemeinen Schutzmechanismen gegen Angriffe.

Martiny

Auflage

Teilgebiet: Textverarbeitung und Büroautomation
edition

1) Anzahl der in Auftrag gegebenen und hergestellten Exemplare eines Druckwerks, ohne Berücksichtigung der notwendigen Druckgänge (→ *Druckaufbereitung*) oder des für den Druck (→ *Druckausgabe*) verwendeten Verfahrens.

2) Ordinalzahl des wiederholten Drucks einer Publikation in größerem zeitlichem Abstand von der vorhergegangenen Drucklegung und nach evtl. vorangegangener Verbesserung oder Erweiterung des Druckgutes.

Ehlers

Auflösungsfehler

Teilgebiet: Analog- und Hybridtechnik
quantization error

Synonyme: Diskretisierungsfehler; Quantisierungsfehler

Ein Fehler aufgrund der Unfähigkeit einer Meßeinrichtung oder eines Datenverarbeitungsgerätes, Veränderungen einer Variablen (→ *Variable*!) wahrzunehmen, die kleiner als ein gegebenes Inkrement (→ *Inkrementgröße*) sind. Bei *analogen* Variablen ist hierfür die Genauigkeit der Meß- oder Verarbeitungseinheit (→ *Processor*) maßgeblich (Toleranzintervall). Bei digital dargestellten Variablen entspricht der Auflösungsfehler der Wertigkeit der niedrigsten dargestellten Ziffer.

Rzehak

Auflösungsprinzip

Teilgebiet: Kognitive Methoden – Künstliche Intelligenz
resolution principle

Auflösungsvermögen

Das Auflösungsprinzip ist eine *logische Schlußregel*, die zwei *Prinzipien* verbindet.

1. Einen Syllogismus des *Aussagenkalküls*:
 $(A \rightarrow B) \wedge (B \rightarrow C) \rightarrow (A \rightarrow C)$
2. Die *Substitutionsregel* des *Prädikatenkalküls*:
 $F(v_1, v_2, v_3, \dots, v_n) \rightarrow F(t_1, t_2, t_3, \dots, t_n)$
Die linke Seite ist eine *Aussage*, die für alle Werte der Variablen $v_1 \dots v_n$ gilt. Auf der rechten Seite sind dafür Terme substituiert, die entweder (andere) Variablen, Konstanten oder beliebig geschachtelte Funktionen von Konstanten sind.

Wenn sich eine Aussage durch geeignete Substitution derart in Faktoren A, B, C zerlegen läßt, daß 1) anwendbar wird, so erhält man eine *logische Folgerung*, die sogenannte *Resolvente* (\rightarrow *Resolventenprinzip*). Zu einer Aussage lassen sich alle Folgerungen nach endlich vielen Anwendungen des Auflösungsprinzips gewinnen. Insbesondere kann nach endlich vielen Schritten ein Widerspruch entdeckt werden, falls die Aussage unerfüllbar ist.

Neumann

Auflösungsvermögen

Teilgebiet: Desktop Publishing

Fähigkeit eines optischen oder elektronischen Gerätes (\rightarrow *Peripheriegerät*) oder Aufzeichnungsmaterials sehr dicht beieinanderliegende Details des wiederzugebenden Objekts noch unterscheidbar abzubilden.

Als Auflösungsstufen werden *Linien/cm* bzw. *Dots per Inch* (dpi) bezeichnet, die neben und untereinander aufgezeichnet werden. *Laserdrucker* zeichnen meist mit 300 bis 600 dpi auf, *Laserbeleuchter* haben oft eine Rasterfeinheit von 2400 dpi bzw. 1000 Linien/cm.

Ehlers

Aufruf

Teilgebiet: Programmierung
call

Mechanismus, der die Ausführung einer vordefinierten *Befehlsfolge* veranlaßt. Diese Befehlsfolge kann als *Makro*, *Unterprogramm* oder als *Hardware*-Funktion vorliegen. Die meisten Maschinen (\rightarrow *Automat*) und *Programmiersprachen* sehen spezielle Befehle bzw. Anweisungen zum Aufruf von Unterprogrammen vor. Im Aufruf können aktuelle *Parameter* definiert werden, die an die Stelle der formalen Parameter in der vordefinierten Befehlsfolge treten.

Würges

Aufruf, abgesetzt

Teilgebiet: Rechnernetze und verteilte Systeme
remote procedure call
Abkürzung: RPC

Eine mächtige Form der Interprozeßkommunikation in *Rechnernetzen*, bei der ein *Sender* prozeß einen Dienst von einem *Server* anfordert, der auf einem anderen *Arbeitsrechner* residiert. Dabei wird ein *Kommunikationsprotokoll* ausgeführt, das die Übertragung der Dienstanforderung und Diensterbringung regelt und dabei evtl. auftretende Übertragungsfehler bzw. partielle Systemausfälle vor dem aufrufenden Prozeß maskieren (\rightarrow *Maske*) soll. Der Sendeprozeß (\rightarrow *Senden*) wird dabei bis zur Ankunft des Ergebnisses der Dienstanforderung blockiert. RPC-Operationen werden zunehmend in verteilten *Betriebssystemen* angeboten.

Bergmann; Leszak

Aufrufbetrieb

Teilgebiet: Datenübertragung
polling mode; selecting mode

Ein Steuerungsverfahren, bei dem innerhalb eines *Datennetzes* eine *Datenstation* aufgerufen wird, zu *senden* oder zu *empfangen*. Die ‚aufrufende‘ Station ist in der Regel die *Leitstation*, die den Betriebsablauf steuert und überwacht.

Eckert

Aufruf eines offenen Unterprogramms \rightarrow *Makrobefehl*

Aufrufsequenz

Teilgebiet: Programmierungsmethodik
calling sequence

Die Instruktion für den *Aufruf* eines *Unterprogramms*. Sie gibt die Aufrufbedingungen (z.B. *Parameter*) an und springt das Unterprogramm an.

Burkhardt

Aufspaltung¹

Teilgebiet: Programmierung
fork

In einem *graphischen Programmablaufplan* eine Stelle oder ein *Knoten*, von der bzw. dem mehrere *Zweige* ausgehen. Dieser Stelle oder diesem Knoten entspricht im *Programmablauf* der Beginn des *parallelen* Ablaufs (\rightarrow *Ablauf eines Programms*) der durch die Zweige dargestellten Teilvorgänge.

Dabei wird begrifflich zwischen *Programmablauf* (als einem Vorgang in der Zeit) und *Programmablaufplan* (als einer Darstellung eines solchen Vorganges) unterschieden. Für die *graphische Darstellung* der verschiedenen Teilvorgänge von *Programmabläufen* gibt es genormte Sinnbilder (\rightarrow *Programmablaufplan*), so auch für ‚Aufspaltung‘ (nach DIN 44 300 T4).

Fleischhauer; Guse; Rouette

Aufspaltung²

Teilgebiet: Programmierung

In einem *Programmablauf* der Übergang von einem Teilvorgang auf mehrere *parallele* Teilvorgänge. Dabei ist es notwendig, zwischen Programmablauf (als einem Vorgang in der Zeit) und *Programmablaufplan* (als einer Beschreibung des Vorganges) zu unterscheiden (nach DIN 44 300 T4). *Fleischhauer; Guse; Rouette*

Auftrag

Teilgebiet: Betriebssysteme

Synonyme: Abschnitt; Bearbeitung

Siehe auch: Datenverarbeitungsauftrag

Eine von einer Person oder Funktionseinheit ausgehende, an eine andere Person oder Funktionseinheit oder an mehrere Personen oder Funktionseinheiten gerichtete, in eine vorab festgelegte Form gefaßte Forderung, eine bestimmte Datenverarbeitungsleistung zu erbringen.

Der Begriff dient der Beschreibung der Beziehungen zwischen Benutzer und Rechensystem (→ *Datenverarbeitungssystem*) wie auch zwischen *Instanzen*. Auftrag soll nicht verstanden werden als Ereignis in der Zeit, sondern als kausale Voraussetzung für das Zustandekommen einer Auftragsbeziehung.

Von entscheidender Bedeutung ist die Kenntnis des organisatorischen Zusammenhangs (des Gesamtsystems), in den das in Anspruch zu nehmende oder zu beschreibende Rechensystem oder die in Anspruch zu nehmende oder zu beschreibende Instanz eingebettet ist. In diesem aus vielen Teilen bestehenden Gesamtsystem muß eine z.T. willkürliche Abgrenzung der Instanz oder des Rechensystems als einer Funktionseinheit gegenüber anderen Funktionseinheiten vorgenommen werden. Ein Rechensystem könnte z.B. als Einheit von Rechenanlage, *Betriebssystem* und Bedienungspersonal verstanden werden.

Daraus ergibt sich eine Schnittstelle, zu deren Lage hier keine Aussage gemacht wird; ihre Festlegung ist jedoch in jedem Einzelfall unerläßlich. Über diese Schnittstelle hinweg werden bestimmte Beziehungen zwischen der Instanz oder der Funktionseinheit Rechensystem und der jeweiligen Umwelt, also den übrigen Funktionseinheiten des Gesamtsystems, gesehen. Daneben treten bei der Beschreibung von Rechensystemen auch Beziehungen in Erscheinung, wie sie zwischen den Teilen des Rechensystems selbst bestehen, u.a. Auftragsbeziehungen.

Durch die Verwendung des Begriffs Auftrag zusammen mit dem Wort „benutzen“ wird deutlich, daß man von der Funktion eines Rechensystems

unterschiedliche Auffassungen haben kann, nämlich

- auf der einen Seite die Funktion eines (Geschäfts-)Partners, dem im Rahmen eines allgemeinen Vertragsverhältnisses Aufträge erteilt werden,
- auf der anderen Seite die eines Werkzeugs, das zur Lösung bestimmter Aufgaben benutzt wird.

Entsprechendes läßt sich auch für in Anspruch zu nehmende Instanzen aussagen. Diese beiden Auffassungen lassen sich durch die Vorstellung miteinander vereinbaren, daß die Rolle des Auftragnehmers teilweise von einer (gegebenenfalls juristischen) Person auf ein Werkzeug übergehen kann. Dieser Übergang geschieht durch Delegation von Aufgaben (nach DIN 44 300 T1 und DIN 66 200).

Beim *Auftragsauswahlverfahren* unterhält jede für die Auftragsvergabe (→ *Auftragsvergabe in Rechnernetzen*) zuständige Instanz eine *Warteschlange* für ankommende Aufträge. Sobald ein *Arbeitsrechner* frei verfügbare Bearbeitungskapazität besitzt, meldet er deren Umfang (z.B. % CPU-Auslastung) der Instanz. Hierzu passend wählt diese dann einen Auftrag aus und weist ihn dem Arbeitsrechner zu. Ein Auftrag besteht häufig aus einer endlichen Menge von möglicherweise gegenseitig abhängigen, in einer *Ordnungsrelation* befindlichen Teilaufträgen. Ein Teilauftrag beschreibt eine im allgemeinen endliche Folge von Funktionen zusammen mit einer zur Ausführung der Funktionen benötigten Menge von *Betriebsmitteln* (z.B. Speicherbedarf).

Ein Auftrag im Dialogbetrieb heißt *Dialogauftrag* oder *Gespräch*, ein Auftrag im *Stapelbetrieb* wird Stapelauftrag oder Abschnitt genannt.

Im Stapelbetrieb resultiert aus einem Auftrag die von einem bestimmten Benutzer zu einem bestimmten Zeitpunkt verursachte Systemlast (→ *Lastprofil*). Ein Auftrag kann gegenüber dem Betriebssystem durch bestimmte Anweisungen der *Betriebssprache* spezifiziert werden. Seine Abarbeitung kann die Initiierung eines oder mehrerer Prozesse und den *Zugriff* auf mehrere *Dateteilen* nach sich ziehen. Aufträge können dem System in einem oder mehreren *Aufgabenströmen* zugeführt werden. Im *Mehrprogrammbetrieb* werden mehrere Aufträge zeitlich verzahnt abgewickelt.

Endres; Fleischhauer; Giloi; Guse; Haupt; Rouette

Auftraggeber

Teilgebiet: Qualitätsmanagement

Kunde in einer Vertragssituation.

Anmerkung: Der Auftraggeber wird manchmal

Auftragnehmer

als die „business second party“ bezeichnet (nach *DIN EN ISO 8402*, Ausgabe August 1995).

Möller

Auftragnehmer

Teilgebiet: Qualitätsmanagement

Lieferant in einer Vertragssituation.

Anmerkungen:

1. Der Auftragnehmer wird manchmal als die „business first party“ bezeichnet.
2. Im Französischen wird der „titulaire du contrat“ gelegentlich „contractant“ genannt (nach *DIN EN ISO 8402*, Ausgabe August 1995).

Möller

Auftragsauswahlverfahren → *Auftragsvergabe in Rechnernetzen*

Auftragsferneingabe

Teilgebiet: Betrieb von Rechensystemen

Eine Betriebsart (→ *Rechensystem, Nutzungsformen eines*), die es erlaubt, Aufträge (Jobs) von einem *Terminal* aus einzugeben. Aus der Sicht des Benutzers gibt es hier keinen Unterschied zu dem gewöhnlichen *Stapelbetrieb*. Die Unterschiede ergeben sich erst in dem *Betriebssystem* selbst.

Diese Betriebsart setzt also ein geeignetes Betriebssystem, Terminalgeräte und *Datenübertragungsleitungen* voraus.

Fialkowski

Auftragsprofil → *Lastprofil*

Auftragsprache → *Kommandosprache*

Auftragsstrom → *Aufgabenstrom*

Auftragsumlaufzeit → *Verweilzeit*

Auftragsvergabe in Rechnernetzen

Teilgebiet: Rechnernetze und verteilte Systeme
network scheduling

Die Auftragsvergabe oder Auftragsdisposition im *Rechnernetz* bewerkstelligt die optimale Zuordnung von Auftrag bzw. Teilauftrag und Bearbeitungssystem (*Arbeitsrechner*) unter Einsatz geeigneter Strategien.

Die Auftragsvergabe umfaßt im einzelnen:

- a) Die Ermittlung der Ausführbarkeit eines Auftrags durch Bestimmung der Menge der zur Auftragsbearbeitung geeigneten Rechner im Rechnernetz.
- b) Die Bestimmung der zu erwartenden „Qualität“ der Auftragsausführung durch die unter a) ermittelten Rechner (z.B. unter Einbeziehung von Informationen über den dynamischen Systemzustand).
- c) Die *Auswahl* des Rechners (→ *Datenverarbeitungsanlage, digitale*) aus der unter a)

ermittelten Menge, der gemäß vorgegebener Optimierungskriterien die optimale Auftragsausführung gewährleistet.

Verfahren der Auftragsvergabe sind Knotenauswahlverfahren, Auftragsauswahlverfahren oder Kombinationen dieser Verfahren. Beim Knotenauswahlverfahren wird jedem Auftrag sofort nach Ermittlung der Ausführbarkeit aus der Menge der zur Ausführung in Frage kommenden Arbeitsrechner derjenige zugeordnet, dessen Einsatz voraussichtlich eine optimale Bearbeitung verspricht. Die Auftragsvergabe kann zentral oder *dezentral* organisiert sein (vgl. *Kontrollstruktur*).

Zitterbart

Auftragszuteilung → *Auftragsvergabe in Rechnernetzen*

Aufwärtskompatibilität → *Baukastenprinzip; kompatibel*

Aufwandschätzung, Methoden der

Teilgebiet: Informatik in der Betriebswirtschaft
cost estimate methods

Unter Methoden zur Aufwandschätzung von DV-Projekten werden im wesentlichen Methoden der Zeit- und Kostenschätzung verstanden. Alle Methoden basieren mehr oder weniger auf den Daten und Erfahrungen abgeschlossener ähnlicher Projekte bzw. Teilprojekte, die sinnvollerweise in einer *Datenbank* gehalten werden sollten (Projektdatenbank, cost data base). Die bekanntesten Methoden - diese Bezeichnung ist vielleicht manchmal zu „methodisch“ - sind die Analogiemethode (Vergleich anhand ähnlicher, abgeschlossener Projekte über Kriterien wie Anwendungsgebiet, *Programmiersprache* etc.), das Multiplikatorverfahren (Basis ist ein Standardmaß wie z.B. die Anzahl realisierter Statements pro Zeiteinheit), das Index-Verfahren (Kriterien wie bei der Analogiemethode werden pro Kriterium in unterschiedliche Intervalle eingeteilt. Diese Intervalle werden dann mit Prozentsätzen bewertet. Ein Beispiel wäre die Bewertung (→ *Informationssysteme, Bewertung von*) der Mitarbeitererfahrung von 3 Jahren mit 100%, von 5 Jahren mit 80% usw.) und das Prozentsatz-Verfahren (die *Phasen* der Systemgestaltung (→ *System'*) werden prozentual gewichtet, z.B. *Vorstudie* mit 10%, *Grobentwurf* (→ *Entwerfen und Konstruieren, rechnerunterstütztes*) mit 15% etc.; nach Fertigstellung oder guter Schätzung der ersten Phase(n) kann der Gesamtaufwand geschätzt werden).

Weitere Methoden sind das IBM-Verfahren, das Aron-Verfahren und das T.O.P.-Verfahren (Terminierung von Organisation und Programmie-

rung). Besondere Bedeutung haben das Constructive Cost Model (Cocomo) von Boehm und die *Function-Point-Methode*. Bei dieser werden die Projekteinheiten wie Eingabedaten, Ausgabedaten, *Datenbestände*, Referenzdaten und Abfragen *klassifiziert* und mit Punkten bewertet. Eine zusätzliche Punktbewertung für die geplante *Realisierung* geht über die sogenannten Einflußfaktoren ein, wie mathematischer Schwierigkeitsgrad, Vernetzung mit bestehenden Anwendungssystemen (\rightarrow *Anwendungssoftware*) etc. Eine auf dieser Weise errechnete Gesamtpunktzahl bewertet das zu realisierende *Projekt*. Anhand einer Regressionskurve realisierter Projekte, bei der die „function-points“ über den Mann-Monaten aufgetragen werden, kann der Aufwand in Mann-Monaten für das neue Projekt „abgelesen“ werden. Die *Nutzung* aller *Verfahren* bedingt die diesbezüglich sorgfältige *Dokumentation* beendeter Projekte und die Beibehaltung des Bewertungsschemas. Bischoff

aufzählbar

Teilgebiet: Berechenbarkeit, rekursive Funktionen
enumerable

Eine Menge (*Relation*) M (Teilmenge von A , $A = N_o^r$ oder $A = (\Sigma^*)^r$, $\Sigma = \text{Alphabet}$, $r \in N_o$) heißt aufzählbar, wenn es einen Algorithmus gibt, der (evtl. mit Wiederholungen) alle Elemente von M erzeugt (aufzählt). Die aufzählbaren Mengen stimmen mit den akzeptierbaren Mengen und den *Definitionsbereichen* (partiell) berechenbarer Funktionen überein. Formale Präzisierung: z.B. Turing-aufzählbar (\rightarrow *Relation*, *Turing-aufzählbare*). Müller

aufzählbare Menge \rightarrow *aufzählbar*

aufzählbare Relation \rightarrow *aufzählbar*

Aufzählbarkeit \rightarrow *aufzählbar*

Aufzählung einer Funktionenklasse \rightarrow *Funktion, universelle*

Augenblickswertverfahren \rightarrow *Analog-Digital-Umsetzung*

Augmented Transition Network

Teilgebiet: Kognitive Methoden – Künstliche Intelligenz
Abkürzung: ATN

ATNs sind Erweiterungen von endlichen Automaten. An den Kanten des Transitionsgraphen eines ATNs können Wörter, beliebige Prozeduren (\rightarrow *Unterprogramm*) und *Aufrufe* von anderen ATNs stehen. Berechnungstheoretisch haben ATNs die Mächtigkeit von *Turing-Maschinen*,

sind also universell. Haupteinsatzgebiet ist das Parsen von Sätzen (\rightarrow *Satz'*) in Fragmenten natürlicher Sprachen. Richter, M.

Ausblenden

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung
reverse clipping; shielding

Das Unterdrücken von *Darstellungselementen*, die innerhalb eines bestimmten Bereichs liegen. Grieger

Ausdruck, arithmetischer \rightarrow *Ausdruck bei Programmiersprachen*

Ausdruck, boolescher \rightarrow *Verzweigungsanweisung (FORTRAN)*

Ausdruck, logischer \rightarrow *Ausdruck bei Programmiersprachen*

Ausdruck, regulärer

Teilgebiet: Automatentheorie und formale Sprachen
regular expression

Zur Beschreibung der von *endlichen Akzeptoren* erkannten (i.allg. nicht endlichen) Wortmengen benutzt man endliche Ausdrücke. Dazu werden Terme über einem Zeichenalphabet gebildet, welches zwei Klassen von Zeichen hat; die erste Klasse besteht aus den Variablen (\rightarrow *Variable'*) A, v_1, \dots, v_n (n natürliche Zahl) und die zweite Klasse aus *Konstanten* (gedeutet als *Operatoren*) $\cdot, \vee, *, +, \wedge, ^c$, sowie dem Klammersymbol $(,)$.

Die Menge der regulären Ausdrücke in n Variablen (Bezeichnung: REG_n) ist durch folgende Forderung rekursiv bestimmt:

- a) Jede Variable bildet einen regulären (\rightarrow *Menge, reguläre*) Ausdruck, also $A \in REG_n$ und $v_i \in REG_n (i=1, \dots, n)$
- b) Wenn $\alpha \in REG_n$ und $\beta \in REG_n$, so auch $(\alpha \cdot \beta), (\alpha \vee \beta), (\alpha)^*, (\alpha + \beta), (\alpha \wedge \beta), (\alpha)^c$
- c) Jeder reguläre Ausdruck ist durch a) und b) bestimmt. Λ ist dabei der leere Ausdruck.

Zwei reguläre Ausdrücke heißen äquivalent, g.d.w. sie dieselbe reguläre Wortmenge beschreiben. Böhling

Ausdruck bei Programmiersprachen

Teilgebiet: Programmierung
expression

Syntaktisch (\rightarrow *Syntax von Programmiersprachen*) korrekte Folge von Zeichen einer Sprache. In *Programmiersprachen* eine Folge von Variablen (\rightarrow *Variable'*), *Konstanten* und Funktionen, die durch *Operationszeichen* verknüpft werden, gemäß der *Syntax* einer Programmiersprache.

Ausfall

Man unterscheidet i.allg. zwischen arithmetischen Ausdrücken, deren Auswertung einen arithmetischen Wert (vom Typ INTEGER (\rightarrow *Integer-Größe*), REAL, COMPLEX usw.) liefert und nur arithmetische Operationen (+, -, \cdot , / usw.) enthält, und *logischen* Ausdrücken, deren Auswertung einen der Werte *true* oder *false* liefert. Ein logischer Ausdruck kann arithmetische Ausdrücke, getrennt durch Vergleichsoperationen (<, \leq , =, \geq , >), enthalten. Würges

Ausfall

Teilgebiete: Grundlegende Definitionen; Rechner-technologie
failure

Eine Abweichung zwischen dem beobachteten und dem beabsichtigten Verhalten eines Systems. Ein Ausfall findet zu einem Zeitpunkt statt, ist also ein Ereignis. Der Ausfall ist auf einen *Fehl-zustand* im System zurückzuführen. Kopetz; Tröller

Ausfall, bösartiger

Teilgebiet: Grundlegende Definitionen
malign failure

Ein *Ausfall*, der möglicherweise katastrophale Folgen (z.B. Gefährdung von Leben) nach sich ziehen kann. Kopetz

Ausfall, gutartiger

Teilgebiet: Grundlegende Definitionen
benign

Ein *Ausfall*, der keine katastrophale Folgen (z.B. Gefährdung von Leben) nach sich ziehen kann. Kopetz

Ausfallabstand, mittlerer

Teilgebiet: Betrieb von Rechensystemen
meantime between failures (MTBF)
Siehe auch: MTBF

Kehrwert der *Ausfallrate* eines Systems, beschreibt über eine längere Betrachtungsdauer den durchschnittlichen zeitlichen Abstand zweier Systemfehler; man kann daraus nicht auf die prozentuale *Verfügbarkeit* des Systems im Betrachtungszeitraum schließen.

Der mittlere Ausfallabstand ist wichtig zur Durchführbarkeit von Aufträgen mit großer *Bearbeitungszeit*. Haupt

Ausfallrate

Teilgebiet: Quantitative Methoden
hazard rate; mortality rate; failure rate
Synonyme: Sterbeintensität; Hazard-Funktion
Ist T die (zufällige) *Lebensdauer* eines Gerätes mit *Verteilungsdichte* $f(t)$, so heißt $\lambda(t) := f(t)/P(T > t)$ die Ausfallrate des Gerätes zur Zeit t . An-

schaulich ist $\lambda(t)$ die Wahrscheinlichkeit dafür, daß das Gerät (z. B. ein Transistor) unmittelbar nach der Zeit t ausfällt, wenn es zur Zeit t noch intakt ist. Bock

Ausfall-Rechenzentrum

Teilgebiet: Betrieb von Rechensystemen
back up processing center
Synonyme: Ausweich-Rechenzentrum; Vorsorge-Rechenzentrum; Back-up-Rechenzentrum

Rechenzentrum, das für einen möglichen Katastrophenfall (\rightarrow *Katastrophenhandbuch*) – ggf. gemeinsam von mehreren Betreibern – vorsorglich bereitgehalten, aber sonst nicht genutzt wird (heiß: voll *betriebsbereit*; warm: nur *Hardware* und Kommunikationseinrichtungen *betriebsbereit*, jedoch nicht Programme und Daten; kalt: *Hardware* und Kommunikationseinrichtungen vorhanden, aber nicht *betriebsbereit*; leer: leerer Raum mit Infrastruktur). Stahlknecht

Ausfallzeit \rightarrow Ausfallzeit, relative

Ausfallzeit, relative

Teilgebiet: Rechner-technologie
down time; fault time

Als (relative) *Ausfallzeit* (*down time*) bezeichnet man den Anteil der planmäßigen *Betriebszeit*, der aus technischen Gründen nicht nutzbar war. Er ist erheblich größer als der Anteil der Reparaturzeiten an der planmäßigen *Betriebszeit*, da bei Auftreten eines Fehlers meist bereits durchlaufene *Betriebszeiten* teilweise nicht mehr nutzbar sind. Das Komplement zur (relativen) *Ausfallzeit* heißt (relative) *Verfügbarkeit* (*availability* (\rightarrow *Verfügbarkeit eines Datenverarbeitungssystems*)). Tröller

Ausführung, direkte

Teilgebiet: Rechnerarchitektur
direct execution

Die direkte Ausführung einer Instruktion der Sprache einer *Architekturebene* erfolgt durch die Erzeugung von *Picoinstruktionen* durch *Hardware-Funktionseinheiten* der *Kontrolleinheit*. Gilot

Ausführung, interpretative

Teilgebiet: Rechnerarchitektur
interpretative execution

Die interpretative Ausführung einer Instruktion einer *Architekturebene* erfolgt durch die Ausführung von gespeicherten Instruktionen der darunterliegenden *Architekturebene*. Gilot

Ausführungsdauer \rightarrow Maschinenzyklus

Ausführungspañwort \rightarrow Paßwort

Ausführungsphase

Teilgebiet: Rechnerorganisation
execution; execution phase

Die Arbeitsphase eines *Rechnerkerns*, in der ein im *Befehlsregister* stehender Befehl ausgeführt wird. In der Ausführungsphase steuert das *Leitwerk* die Abläufe (*→ Ablauf eines Programms*) im *Rechenwerk* bzw. Ein-/Ausgabewerk. *Klar*

Ausführungszeit *→ Antwortzeit²*

Ausführungszeit eines Unterbrechungsantwortprogramms

Teilgebiet: Prozeßdatenverarbeitung

Die *Ausführungszeit* umfaßt nur die Ablaufzeit des *unterbrechungsspezifischen Antwortprogramms*. Die Identifizierung der Unterbrechungsursache und die Bereitstellung der Betriebsmittel werden nicht eingerechnet und bei der *Erkennungszeit* des *Unterbrechungssignals* eingeschlossen. Bild siehe bei: Antwortzeit (*→ Antwortzeit²*). *Lauber*

Ausgabe

Teilgebiet: Allgemeines

Ergebnis eines Programmablaufes, in einer vom Menschen interpretierbaren Form auf einem *Bildschirm dargestellt* oder ausgedruckt, in einer *Datei* abgespeichert oder über eine Übertragungsleitung (*→ Verbindungsleitung*) geschickt. *Nowak*

Ausgabe, akustische

Teilgebiet: Interaktive Systeme, Mensch-Maschine-Dialog

Ausgabe in hörbarer Form, als Klick, Summton o.ä. oder als Sprachausgabe. *Nowak*

Ausgabe, digitale

Teilgebiet: Allgemeines

Ausgabe eines Rechners (*→ Datenverarbeitungsanlage, digitale*) in Form von Codezeichen (als elektrische Signale), die dann in einem entsprechenden *Ausgabegerät* in eine vom Menschen lesbare Form umgewandelt, in einer *Datei* abgespeichert oder entfernt übertragen werden. *Nowak*

Ausgabe, graphische

Teilgebiet: Interaktive Systeme, Mensch-Maschine-Dialog

Ausgabe in Form von Zeichnungen oder Graphiken auf einem *Bildschirm* oder *Drucker*. *Nowak*

Ausgabe, lokale

Teilgebiet: Rechnerperipherie

Ausgabe auf einem *Ausgabegerät* (Bildschirm, Drucker oder Speichermedium wie Magnetplatte

(*→ Magnetplattenspeicher*) oder Magnetband (*→ Magnetbandspeicher*)), das direkt an den Rechner angeschlossen ist. *Nowak*

Ausgabealphabet *→ Kellerautomat*

Ausgabeband *→ Kellerautomat*

Ausgabebefehl

Teilgebiete: Programmierung; Rechnerperipherie

Siehe auch: Ein-/Ausgabebefehl

Befehl (einer *Programmiersprache* – z.B. *Maschinensprache*) zur Übertragung von Daten an ein externes, an den Rechner angeschlossenenes Gerät. Die *Parameter* (bzw. *Operanden*) des Befehls bestimmen das gewünschte Gerät, die zu übertragende *Nachricht* und evtl. Steueranweisungen (*→ Steuerbefehl*) für das Gerät. *Würges*

Ausgabedatei

Teilgebiet: Allgemeines

Datei im *Hauptspeicher* oder auf einem Nahperipheriegerät, die eine *Ausgabe* zur *Archivierung* oder zur nachfolgenden *Darstellung* bzw. *Fernübertragung* aufnimmt. *Nowak*

Ausgabereinheit

Teilgebiet: Rechnerorganisation

Eine Funktionseinheit innerhalb eines *digitalen Rechensystems*, mit der das System Daten, z.B. Rechenergebnisse, nach außen hin abgibt. *Schneider*

Ausgabegerät

Teilgebiet: Rechnerorganisation
output device

In einer *Ausgabereinheit* eine *Baueinheit*, durch die Daten aus einer Rechenanlage ausgegeben werden können. *Schneider*

Ausgangsdaten *→ Basisdaten*

Ausgangsfunktion

Teilgebiet: Berechenbarkeit, rekursive Funktionen
base function

Synonyme: Basisfunktion; Grundfunktion

Siehe auch: Funktion, primitiv-rekursive

Die Klassen der elementaren, primitiv-rekursiven (*→ primitive Rekursion*) und rekursiven Funktionen werden induktiv (*→ Induktion*) definiert, aufbauend jeweils auf einer Menge von (als besonders einfach angesehenen) Funktionen, die man Ausgangs-, Grund- oder Basisfunktionen nennt. Dazu gehören die *konstanten Funktionen* und *Projektionsfunktionen*. *Müller*

Ausgangsleitung → *Wegeermittlung; Chip-select*

Ausgangsspannung

Teilgebiet: Schaltwerke und Schaltnetze
output signal; output voltage

Spannung am Ausgang einer elektrischen Schaltung. Der Begriff „Ausgang“ ist dabei nach DIN 41859 definiert als das „Anschlußpaar (Anm.: einer Digitalschaltung), dem digitale elektrische Ausgangsgrößen entnommen werden“.

Diese für Digitalschaltungen formulierte *Definition* gilt sinngemäß auch für nichtdigitale Schaltungen, beispielsweise für *Funktionsgeneratoren*.
Schmid, D.

Ausgangsstufe → *Endstufe*

Ausgleich → *Saldenübereinstimmung*

Ausgleichen

Teilgebiet: Desktop Publishing
kerning

Beim Handsatz wurden durch *Spationieren* und *Unterschneiden* besonders bei Versalzzeichen die durch die jeweilige Buchstabenkombination sich ergebenden optisch unterschiedlichen Zwischenräume (z.B. PA, AV, WT, Te, To etc.) ausgeglichen. Entsprechende *Ästhetikprogramme* können diese Aufgaben heute *software* gesteuert erledigen.
Ehlers

Auskunftssystem

Teilgebiet: Grundlagen von Informationssystemen

Ein Auskunftssystem dient der Wiedergabe gespeicherter Daten. Im allgemeinen geschieht dies mittels wahlfreiem *Zugriff* auf Informationen, die in einer *Datenbank* gespeichert sind, mit Hilfe einer problemorientierten Sprache über *Terminals*. Ein Auskunftssystem ist z.B. meist Bestandteil eines *Managementinformationssystems*.
Meyer, B.

Auslastung

Teilgebiete: Betrieb von Rechensystemen; Rechnerarchitektur

relative throughput; utilisation

Synonym: Durchsatz, relativer

Unter Auslastung eines *DV-Systems* oder einer Funktionseinheit oder einer Systemkomponente versteht man das Verhältnis von der in einem Zeitraum erbrachten *Leistung* zu der in diesem Zeitraum maximal erbringbaren Leistung (*peak performance* (→ *Verkehrsgüte*)). Da die Auslastung eines *DV-Systems* nicht nur von der Menge und der Größe der zu bearbeitenden Program-

me, sondern auch von charakteristischen Eigenschaften der zu bearbeitenden Aufgaben abhängt, ist es nahezu unmöglich, eine allgemeine maximale Auslastung zu definieren. Man zieht daher Vergleichsläufe mit für die Aufgabenstellung charakteristischen Programmen, sogenannten *Benchmarks*, heran. Die Auslastung eines *parallel* bzw. massiv parallel arbeitenden Systems durch ein einziges Programm hängt sehr davon ab, wie weit die innere Struktur der Aufgabenstellung eine Parallelisierung erlaubt. Einfacher ist die Definition der Auslastung einzelner Komponenten des *DV-Systems*, da man sich hier leichter auf nominale Nennwerte beziehen kann. Weiterverbreitet ist die beschränkte Betrachtung der *CPU-Auslastung* oder der *Kanalauslastung*. Neben der Auslastung der aktiven *Betriebsmittel*, der *Prozessoren* (*Prozessorauslastung*) und *Transportkanäle* (→ *Kanal*) (*Kanalauslastung*) gibt die Auslastung der passiven *Betriebsmittel*, des *Hauptspeichers* und der *Hintergrundspeicher* (*Speicherauslastung*) sowie der Geräte (→ *Peripheriegerät*) (*Geräteauslastung*) wichtige Hinweise darauf, ob das *DV-System* richtig *konfiguriert* wurde.

Die entsprechenden *Belegungszeiten* (→ *Belegung eines DV-Systems*) werden vom *Accounting-System* (→ *Abrechnungsroutine*) registriert.
Ameling; Beilner; Graef; Haup

Auslöschung → *Zahldarstellung*

Ausnahmebedingung → *ON-Bedingung*

Ausnahmebehandlung

Teilgebiete: Allgemeines; Programmierungsmethodik

exception handling

Die Ausnahmebehandlung kommt in Situationen zum Tragen, in denen man zur *Implementierungs- und Planungszeit* noch nicht sämtliche Möglichkeiten der *Datenausprägungen* kennt.

Systeme mit der Möglichkeit der Ausnahmebehandlung lassen während des Laufs eine *Instruktion* des Systems zu. Realisiert wird die Ausnahmebehandlung durch *Retten* von Anwendersystemzuständen, durch *Verlassen* des Anwendungssystems (→ *Anwendungssoftware*), durch anschließendes *Arbeiten auf Betriebssystemebene* und durch definiertes *Zurückkommen* in das *Anwenderprogramm* (→ *Anwendungsprogramm*) sowie *Fortfahren* am *Unterbrechungspunkt*.
Richter, J.; Schneider

Ausnahmelexikon → *Silbentrennung*

Ausprägung (einer Daten-Bauart)

Teilgebiet: Datenstrukturen und Datenoperationen

Digitale Daten, die einer *Daten-Bauart* genügen, also einer Vorschrift über den Aufbau aus Zeichen oder deren Anordnung. Dabei können je zwei Ausprägungen derselben Daten-Bauart in allem übereinstimmen oder nicht; dementsprechend heißen sie gleiche bzw. unterschiedliche Ausprägungen (nach *DIN 44 300 T3*).

Fleischhauer; Rouette

Aussage

Teilgebiet: Logik
statement

- a) Aussage eines Satzes (→ *Satz*¹): gedankliches Abbild eines Sachverhalts.
- b) Ein Aussage-Satz ist ein sprachliches Gebilde, das einerseits nach gewissen *syntaktischen Regeln* (z.B. Subjekt – Prädikat – Objekt) gebildet ist, und dem andererseits eine der Eigenschaften wahr oder falsch zukommt.
- c) (formal) Eine Aussage ist eine Formel ohne freie Variablen (→ *Variable*¹).

Müller

Aussage, elementare → *Beziehung, elementare*

Aussage, prädikatenlogische → *Semantik, axiomatische*

Aussageform

Teilgebiet: Logik
sentential form

Sprachliches Gebilde, das Variablen (→ *Variable*¹) enthält und durch *Ersetzung* aller Variablen durch zulässige Objektamen in eine *Aussage* übergeht. In der formalen *Logik* entspricht „Aussageform“ der Terminus *Formel*.

Müller

Aussagenkalkül

Teilgebiet: Logik
sentential calculus

Siehe auch: Aussagenlogik; Logikkalkül

Logikkalküle der Aussagenlogik werden als Aussagenkalküle bezeichnet. Aussagenkalküle können sich in Bezug auf die zugrundeliegenden *Axiome* unterscheiden. Bekannte, zueinander äquivalente *Kalküle* gehen u.a. auf Frege, Whitehead/Russell, Hilbert/Ackermann und Lukasiewicz zurück. Außerdem ist der intuitionistische Kalkül Heytings von großer Bedeutung.

Habel

Aussagenlogik

Teilgebiet: Logik
assertion logic; propositional calculus
Synonym: Junktorenlogik

Aussagenlogik ist der Teil der (formalen) *Logik*, der sich mit *Junktoren* (aussagenlogische Ver-

knüpfungen) und den damit aufgebauten Formeln beschäftigt. Sie liefert Aussagen über den *Wahrheitswert* einer Formel, die nur von den Wahrheitswerten der aussagenlogischen Bestandteile abhängen.

Müller

aussagenlogisch wahr → *wahr, aussagenlogisch*

aussagenlogisch widerspruchsvoll → *kontradiktorisch*

Ausschaltseinheit → *Funktionseinheit*

Ausschießen

Teilgebiet: Desktop Publishing
imposition

Zusammenstellen der Druckformen, damit nach dem Vorder- und Rückseitendruck und Falzen des mehrere Seiten umfassenden Druckbogens die Seiten in der richtigen *Reihenfolge* erscheinen. Beim digitalen Druck (→ *Druckausgabe*) ist eine automatische Ausschießsoftware unabdingbar.

Ehlers

Ausschließen

Teilgebiet: Textverarbeitung und Büroautomation
justified setting

In der Satzherstellung werden durch Ausschließen die Wortzwischenräume einer *Zeile* so geändert, daß sich eine festgelegte Breite für jede Zeile ergibt. Nicht ausgeschlossener Satz wird als *Flattersatz* bezeichnet.

Ehlers

Ausschluß, wechselseitiger → *Monitor, Hoares*

Ausschreibung

Teilgebiet: Einsatzplanung und Installation
competition; submission

Trotz entsprechender Vorschriften in der *VOL* kann die Ausschreibung als Mittel zur Ermittlung des Lieferanten einer *DV-Dienstleistung* nur sehr eingeschränkt angewendet werden. Der Grund hierfür liegt in der Tatsache, daß sich die benötigte *Leistung* nach Art und Umfang nicht exakt genug formulieren läßt und darüber hinaus auch die durch Verbesserung von *Software* und Ergänzungen der *Hardware* erhofften Leistungsverbesserungen (→ *Leistungsermittlung*) bei der *Auswahl* des Lieferanten wesentlich sind. In den meisten Fällen wird allen infrage kommenden Lieferanten eine für alle gleiche *Liste* von Fragen und *Anforderungen* zugestellt, der Vergabevorgang aber nicht nur unter Preisaspekten vorgenommen.

Haupt

Außensperre

Teilgebiet: Prozeßdatenverarbeitung

Die Außensperre ist ein Teil der Unterbrechungs-Eingabeeinheit und dient zum (gezielten) Aussperren von *Unterbrechungssignalen*. Beispiel in DIN 66216 Blatt 2 Seite 3. Näheres siehe bei Unterbrechungs-Unterdrückung.

Lauber

Austasten

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

blinking

Siehe auch: Clippen

Das *Löschen* von *Darstellungselementen* oder Anliegegruppen, die ganz oder teilweise außerhalb eines Fensters liegen.

Encarnação

Austastlücke

Teilgebiet: Grundlegende Definitionen

Die Leerzeilen der 625 Bildzeilen in der PAL-Fernsehnorm, die die Bildröhre während des Bildwechsels dunkel steuern, im *Videotext* heute auch häufig als Fernsehtext bezeichnet, werden zur Übertragung der Seiten benutzt. In der Zeitspanne, in der die nicht mit Bildinhalt belegten Zeichen und Synchronisierungsimpulse für den Bildwechsel durchlaufen, wird die Bildröhre dunkel gesteuert, d.h. ausgetastet und dabei die Videotext-Information im „Huckepack-Verfahren“ übertragen.

Fuchs

Austausch, paarweiser → *Sortieren durch Austausch*

Austauschbarkeit

Teilgebiet: Grundlegende Definitionen

Fähigkeit eines Erzeugnisses eines Verfahrens oder einer *Dienstleistung*, anstelle eines (einer) anderen benützt zu werden, um dieselben *Anforderungen* zu erfüllen.

Anmerkung: Der funktionelle Aspekt der Austauschbarkeit wird „funktionelle Austauschbarkeit“ und der Aspekt der Abmessungen „dimensionelle Austauschbarkeit“ genannt (nach *DIN EN 45 020, Ausgabe April 1994*).

Möller

Austauschformat → *ODA*

Austausch nach Auswahl → *Sortieren durch Auswahl*

Auswähler

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

choice device

Eingabegerät, das einen Wert liefert, der aus einer Menge von Alternativen ausgewählt wird. Die typische *Realisierung* dieses Eingabegeräts sind die Funktionstasten.

Grieger

Auswahl → *Selektion*

Auswahlantwort

Teilgebiet: Computergestützter Unterricht und Pädagogik

multiple choice

Lehr- und Übungsstrategien des Computerunterstützten Unterrichts (CUU (→ *Unterricht, computerunterstützter*)) bieten gleichzeitig eine Frage mit mehreren möglichen Lösungen an. Der Adressat wählt aus dem Lösungsangebot die seiner Meinung nach richtige(n) Lösung(en) aus und übermittelt seine *Auswahl* mit Hilfe einer *Maus*, eines *Lichtgriffels* oder des *Cursors*, manchmal auch durch die Eingabe des (der) Kennzeichen der gewählten Lösungen an das Lehrprogramm (→ *Lehrprogramme, Klassifikation von*). Dieses reagiert auf die Antwort des Adressaten in der vom Autor vorgegebenen Weise.

Gunzenhäuser

Ausweich-Rechenzentrum → *Ausfall-Rechenzentrum*

Auswertung, partielle

Teilgebiete: Programmierung; Übersetzerbau

partial evaluation

Synonym: Vorberechnung, mixed computation

Unter partieller Auswertung einer Funktion $f: A \times B \rightarrow C$ versteht man die auf der mathematischen *Äquivalenz* $f: A \times B \rightarrow C \equiv f': A \rightarrow (B \rightarrow C)$ beruhende Auswertung der Funktion f' , die für alle möglichen Argumente $a \in A$ spezielle Funktionen $f'_a: B \rightarrow C$ ergibt. Auf diese Weise erhält man etwa aus der Multiplikation $\text{mult}: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ durch partielle Auswertung für das (feste) erste Argument 2 eine Funktion $\text{dup}: \mathbb{N} \rightarrow \mathbb{N}$ für das Verdoppeln. Ein Berechnungsvorteil durch Anwendung der partiellen Auswertung ergibt sich insbesondere dann, wenn ein Argument einer (häufig angewandten) Funktion f nur wenige aktuelle Werte annehmen kann. In diesem Fall führt nämlich die *Ersetzung* aller Funktionsanwendungen von f durch die (durch partielle Auswertung erhaltenen) speziellen Funktionen zu einem deutlich geringeren Berechnungsaufwand.

Parsch

Auswertung von Daten → *Datenanalyse*

Auszeichnen

Teilgebiet: Textverarbeitung und Büroautomation

typographic styling

In der Satzherstellung *Hervorheben* von Textteilen durch Unterstreichen, *Sperren* oder andere Schriftgrade oder -schnitte. Farbe oder Negativdarstellung (→ *Negation*) werden ebenfalls zur *Auszeichnung* verwendet.

Ehlers

Auszeichnung → *Satz, glatter*

AUT

Teilgebiet: Informatik im Bankwesen

Zugriffsbedingung, die durch ein *Challenge-Response-Verfahren* erfüllt wird. Martiny

Authentifikation

Teilgebiet: Betriebssysteme
authentication

Die vom *Datenverarbeitungssystem* veranlaßte *Überprüfung* der Identität eines Benutzers (Person, Computer, Programm), der Zugang zum System (*Zugriff* auf Daten und Programme) haben will.

Je nach den Sicherheits- und Datenschutzeinrichtungen des Systems kann diese Überprüfung viele verschiedene Formen annehmen. Beispiele dafür sind *Paßworte*, *Magnetcodes* (Badges) oder ein identifizierender Dialog zwischen System und Benutzer. Mätzel; Neuhold

Authentifikation, externe

Teilgebiet: Informatik im Bankwesen

Nachweis der Echtheit der Anwendungsumgebung gegenüber der *Chipkarte*. Martiny

Authentifikation, interne

Teilgebiet: Informatik im Bankwesen

Nachweis der *Berechtigung* des Endgeräts gegenüber der *Chipkarte*. Martiny

authentisch

Teilgebiet: Informatik im Bankwesen

Vorgegebener Ursprung der *Nachricht* (oder des Gegenstands) ist echt. *Nachweis* der Echtheit wird als Authentifikation bezeichnet. Martiny

Autobackout

Teilgebiet: Grundlagen von Informationssystemen

Zu Beginn einer Datenbanksitzung werden automatisch alle im Sinne der Transaktionslogik (*Transaktion*) noch nicht abgeschlossenen *Datenbank* veränderungen zurückgesetzt (z.B. nach ungewolltem *Systemabbruch*). Freiburg

Autodekrement-Adressierung

Teilgebiet: Programmierung
autodecrement addressing

Eine Variante der *Adreßbestimmung* für indizierte (bzw. indirekte) Adressen, bei der (i.allg.) nach der Adreßbestimmung der elementaren Adresse das an der Adreßbestimmung beteiligte *Register* (i.allg.) um eins (oder um den Inhalt eines zweiten Registers) erniedrigt wird. Beim Erreichen des Wertes Null erfolgt eine Markierung, die beispielsweise zum *Abbruch* einer Programmschlei-

fe (→ *Schleife*¹) führen kann (siehe auch *lineare Adressenfortschaltung*). Hoffmann, H.-J.

Autoindizierung

Teilgebiet: Rechnerorganisation
autoindexing

Um *Indexregister* einzusparen, hat man bei manchen frühen Rechnern (→ *Datenverarbeitungsanlage, digitale*) einige Zellen (→ *Speicherzelle*) des Arbeitsspeichers (→ *Hauptspeicher*) (MEM) mit der Fähigkeit ausgestattet, den Zelleninhalt bei jedem Lesevorgang (→ *Lesen*) automatisch zu inkrementieren (→ *Inkrementgröße*):

Lesen:

(Speicherdatenregister) ← MEM ((Adr))

Inkr.:falls die gewählte Adresse auf eine selbstindizierende Speicherzelle zeigt, erfolgt zusätzlich ohne weiteren Befehl oder weitere *Verzögerung*:

MEM ((Adr)) ← MEM ((Adr)) + 1.

Dieser Mechanismus wird Autoindizierung genannt. Klar

Autoinkrement-Adressierung

Teilgebiet: Programmierung
autoincrement addressing

Eine Variante der *Adreßbestimmung, analog* zur *Autodekrement-Adressierung*. Das an der Adreßbestimmung beteiligte *Register* wird erhöht; Markierung erfolgt, wenn ein (z.B. in einem weiteren Register) voreingestellter Wert erreicht wird (siehe auch *Autoindizierung*). Hoffmann, H.-J.

Autokorrelation

Teilgebiet: Quantitative Methoden

Korrelation zwischen den Größen X (t) und X (t+Δt) eines Zufallsprozesses. Stahl

Automat

Teilgebiet: Automatentheorie und formale Sprachen

automaton

Synonym: Maschine

Modellmechanismus (→ *Gegenstandsraum*) zur *Verarbeitung* von *Datenstrukturen*. Elemente einer abstrakten Eingabemenge *I* werden unter speichergestützter innerer *Kontrolle* abgearbeitet, transformiert oder erkannt und zur weiteren Verwertung in einer Ausgabemenge *O* ausgegeben.

$$I \rightarrow \boxed{A} \rightarrow O$$

Man erfaßt das Verhalten eines Automaten *A* mit abstrakten Zuständen einer Zustandsmenge *Q* über eine Transitionsfunktion

$$\tau_A: Q \times I \rightarrow Q \times O$$

In der *Automatentheorie* werden Arbeitsweise und Eigenschaften von Automaten bei verschiedenen Strukturannahmen über die Grundmengen I, O, Q untersucht.

Bei Erfassung (\rightarrow *Datenerfassung*) spezieller Datenstrukturen sind geeignete Annahmen zu machen über Ein- und Ausgabeelemente. Z.B. wird bei Verarbeitung von *Zeichenketten* eine endliche Zeichenmenge Z (*Alphabet*) zugrunde gelegt und die Menge aller endlichen Zeichenketten als freies Monoid Z^* (Halbgruppe mit Einselement e) betrachtet; bei nichtlinearen Datenstrukturen sind z.B. *Bäume* Elemente der Eingabemenge.

Die Zustandsmenge Q beschreibt Eigenschaften einer *Kontrollstruktur* und die Wirkungsweise von Arbeitsspeichern über geometrisch strukturierte Speichermedien samt notwendiger Hilfspesichersymbolik. Die Kontrollstruktur wird meist als endliche Menge (Alphabet) S von internen Zuständen angenommen.

Eine Menge von *Konfigurationen* eines Automaten beschreibt die einzelnen Schritte der *Informationsverarbeitung* unter den getroffenen Strukturvoraussetzungen. Ausgehend von einer *Anfangskonfiguration*, in der ein Anfangszustand und ein Anfangsargument der Eingabemenge festgelegt werden, verarbeitet ein Automat A schrittweise dieses *Eingabeelement* entsprechend seiner inneren Struktur durch Änderung interner Zustände, bis eine Endkonfiguration erreicht wird, bei der A in einem Endzustand aus einer Teilmenge *ausgezeichneter Zustände* anhält und das Verarbeitungsergebnis als Element der Ausgabemenge liefert.

Ein Automat wird unter Erfassung von Anfangs- und Endbedingungen als 6-Tupel (\rightarrow *n-Tupel*) definiert:

$$A = (Q, I, O, \tau_A, q_0, F)$$

mit der Zustandsmenge Q (einschließlich der Speicherzustände), dem Anfangszustand q_0 , der Menge $F \subseteq Q$ der Endzustände, der Eingabemenge I , der Ausgabemenge O und der Transitionsfunktion τ_A .

Läßt man Anfangs- und Endbedingungen außer acht und faßt nur die Grundmengen sowie die das Verhalten beschreibende Transitionsfunktion τ zusammen, so spricht man von einer Maschine

$$M = (Q, I, O, \tau)$$

In der Fachliteratur wird häufig das Tupel M bereits als Automat bezeichnet.

Deterministische Automaten haben ein eindeutig bestimmtes Transitionsverhalten (τ_A ist eine Abbildung). Bei nichtdeterministischen Automaten ist die Bildzuordnung (d.h. nächster Zustand

bzw. *Ausgabe*) nicht eindeutig festgelegt, und die Beschreibung der Beziehungen erfolgt durch eine *Relation* $\tau_A \subseteq Q \times I \times Q \times O$ oder durch eine Abbildung in die Potenzmenge (Menge aller Teilmengen) des Bildbereichs (\rightarrow *Darstellungsbereich*) $\tau_A: Q \times I \rightarrow POT(Q \times O)$.

Erfolgt die Transition mit einer gewissen *Wahrscheinlichkeit*, so wird das Verhalten durch bedingte Wahrscheinlichkeiten beschrieben (*stochastische Automaten*). Automaten mit leerer Ausgabemenge $O = \emptyset$ werden als Akzeptoren (\rightarrow *Akzeptor, endlicher*) bezeichnet.

Automaten ohne Arbeitsspeicher (\rightarrow *Hauptspeicher*), d.h. $Q = S$, werden bei endlichen Automaten betrachtet.

Automaten mit spezieller Struktur des Arbeitsspeichers sind z.B. die *Turing-Maschine* oder der *Kellerautomat*.

Den Automaten verwandte Strukturen sind *abstrakte Maschinen*, insbesondere *Registermaschinen* als idealisierte Modelle für Rechenmaschinen.

Böhling

Automat, abelscher zellularer \rightarrow *Automat, zellularer*

Automat, autonomer \rightarrow *Automat, endlicher*

Automat, deterministischer \rightarrow *Automat, endlicher*

Automat, endlicher

Teilgebiet: Automatentheorie und formale Sprachen

finite state machine

Synonym: Automat, initialer

Ein endlicher *Automat* ist ein Automat mit endlicher *Kontrollstruktur* ohne Arbeitsspeicher zur Beschreibung der *Verarbeitung* von *Zeichenketten* (Erkennung, Erzeugung, Transformation)

$$A = (S, X, Y, \tau_A, s_0, F)$$

mit dem Zustandsalphabet S

dem *Eingabealphabet* X

dem *Ausgabealphabet* Y

der Transitionsfunktion

$$\tau_A: S \times X \rightarrow S \times Y$$

dem Anfangszustand s_0

und der Menge der Endzustände $F \subseteq S$

$\tau_A(s, x) = (s', y)$ bedeutet, der Automat A empfängt im Zustand s das Eingabezeichen x , geht dann in den Zustand s' über und gibt dabei das Ausgabezeichen y aus.

Zerlegt man die Transitionsfunktion τ_A in eine Überföhrungsfunktion (\rightarrow *Übergangsfunktion*)

$$\delta: S \times X \rightarrow S$$

$$\text{und eine Ergebnisfunktion}$$

$$\lambda: S \times X \rightarrow Y,$$

so bezeichnet man A als *MEALY-Automat*, ist die Ergebnisfunktion eingabeunabhängig $\mu: S \rightarrow Y$, so heißt A *Moore-Automat*.

Endliche Automaten vermitteln Automatentransformationen von Zeichenketten $\Phi: X^* \rightarrow Y^*$, indem schrittweise (*sequentiell*) die Zeichen eines Eingabewortes $u = u_1 \dots u_k$ werden über Zustandsänderungen eines Automaten A in Ausgabezeichen umgeformt

$$s_0, s_1 = \delta(u_1, s_0), \dots, s_{i+1} = \delta(u_{i+1}, s_i) \quad (i=0, \dots, k-1)$$

$$s_k = \delta(u_k, s_{k-1}) \in F \text{ Kürzer: } s_k = \delta^*(u, s_0)$$

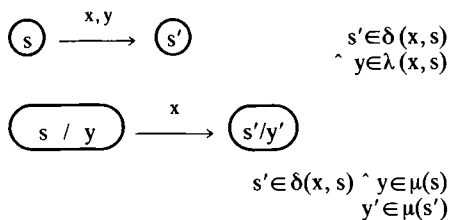
$$v_1 = \lambda(u_1, s_0), \dots, v_k = \lambda(u_k, s_{k-1}), \quad v = \lambda^*(u, s_0)$$

$$v = v_1 \dots v_k \text{ Ausgabewort.}$$

Ist das Eingabealphabet einelementig (d.h. enthält es nur ein synchronisierendes Taktsignal (\rightarrow *Taktimpuls*)), so spricht man von autonomen Automaten. Je nachdem, ob die Transitionsfunktion eindeutig oder mehrdeutig ist, bezeichnet man A als deterministischen oder nichtdeterministischen endlichen Automaten.

Beim nichtdeterministischen Automaten ist nichts über *Prioritäten* oder *Wahrscheinlichkeiten* für das Auftreten der Nachfolgezustände (\rightarrow *Nachfolgerfunktion*) gesagt.

Das *Transitionsverhalten* von Automaten kann veranschaulicht werden durch gerichtete *Graphen*, deren *Knoten* durch Zustände markiert werden (Zustandsdiagramm (\rightarrow *Zustandstabelle*)) und deren gerichtete Kanten mit Eingabezeichen markiert werden. Je nach MEALY- oder MOORE-Fall können die Ausgabezeichen an den Kanten bzw. Knoten markiert werden.



Anfangszustand und Endzustände können besonders markiert werden. Ein endlicher Automat heißt schwach zusammenhängend, wenn das Zustandsdiagramm zusammenhängend ist, d.h. nicht in *Teilgraphen* zerfällt, stark zusammenhängend, wenn jeder Zustand vom Anfangszustand aus erreichbar ist. Böhling

Automat, endlicher erkennender \rightarrow *Akzeptor, endlicher*

Automat, erweiterter endlicher
 Teilgebiete: Automatentheorie und formale Sprachen; Nebenläufige Prozesse
extended finite-state machine
 Abkürzung: EFSM

Ein erweiterter *endlicher Automat* entsteht aus einem endlichen Automaten, indem die Zustände

des *Automaten* durch Elemente eines (evtl. unendlichen) *Datentyps* parametrisiert (\rightarrow *Parameter*) werden. Die *Belegung* eines Folgezustands mit Werten für die Parameter und die Frage, ob eine Aktion in einem Zustand des erweiterten endlichen Automaten aktiviert ist, hängt dann von der Belegung dieses Zustands ab.

Erweiterte endliche Automaten werden als *Referenzmodell* für einige *Spezifikationssprachen* für verteilte Systeme benutzt. Bergmann; Leszak

Automat, initialer \rightarrow *Automat, endlicher*

Automat, linear beschränkter

Teilgebiet: Automatentheorie und formale Sprachen
linear bounded Turing machine

Restriktive *Turing-Maschine*, bei der die Länge des Speicherbandes für die Berechnung eines Resultats durch die Länge des Arguments begrenzt ist. Dazu verwendet man bei Einband-Turing-Maschinen zusätzliche Bandmarken, die das Eingabewort beidseitig begrenzen und während des Verarbeitungsprozesses (\rightarrow *Prozeß*²) nicht überschritten werden dürfen.

Bei Mehrband-Turingmaschinen ist der gesamte Speicherbedarf aller Bänder durch eine lineare Funktion der Länge des Eingabewortes nach oben beschränkt (Sonderfall der Bandkomplexität bei Turingmaschinen).

Nicht-deterministische linear beschränkte Turing-Akzeptoren erkennen genau die kontextsensitiven Sprachen. Böhling

Automat, linearer

Teilgebiet: Automatentheorie und formale Sprachen
linear sequential machine

Sind die Bestimmungsstücke S, X, Y eines *sequentiellen Systems* $M = (S, X, Y, \delta, \lambda)$ lineare Räume über einem endlichen Körper (z.B. arithmetische Vektorräume), so lassen sich Überföhrungsfunktion δ und Ergebnissfunktion λ durch lineare Beziehungen darstellen:

$$\delta(\underline{s}, \underline{x}) = A\underline{s} + B\underline{x} = \underline{s}'$$

$$\lambda(\underline{s}, \underline{x}) = C\underline{s} + D\underline{x} = \underline{y}$$

$\underline{s}, \underline{y}, \underline{s}, \underline{s}'$ sind Vektoren aus den entsprechenden linearen Räumen und A, B, C, D die das System M charakterisierenden Matrizen von mit den Räumen verträglicher Dimension.

A beschreibt die autonomen Transitionseigenschaften von M . Als linearer *Automat* wird dann das Tupel (\rightarrow *n-Tupel*) $[A, B, C, D]$ der charakterisierenden Matrizen bezeichnet.

Ein linearer Automat heißt nicht-singulär, falls die Matrix A nicht singulär ist. Böhling

Automat, minimaler

Automat, minimaler → Automaten, Reduktion endlicher

Automat, nichtdeterministischer → Automat, endlicher

Automat, reduzierter → Automaten, Reduktion endlicher

Automat, schwach zusammenhängender → Automat, endlicher

Automat, sequentieller → System, sequentielles

Automat, stark zusammenhängender → Automat, endlicher

Automat, stochastischer

Teilgebiet: Automatentheorie und formale Sprachen
probabilistic automaton

Bei Automaten, die Zeichenketten (Wörter) verarbeiten, kann die Transitionsfunktion durch bedingte *Wahrscheinlichkeiten* für die Zustandsänderung und Zeichentransformation gekennzeichnet sein.

Dann bezeichnet man solche Automaten als stochastische Automaten. Die Eingabemenge X , Ausgabemenge Y und Zustandsmenge S sind höchstens abzählbar (im Falle endlicher Mengen heißt der Automat endlich). Die Transitionsfunktion $\tau_A: S \times X \rightarrow S \times Y$ wird durch Angabe von bedingten Wahrscheinlichkeiten $p(s', y; s, x)$ für jeden Übergang vom Zustand s bei Eingabe des Zeichens x in den nachfolgenden Zustand s' unter Ausgabe des Zeichens y beschrieben.

Zweckmäßig fällt man bei endlichen *stochastischen* Automaten diese *Übergangswahrscheinlichkeiten* in einem Matrixensystem

$$W = (W(x, y))_{x \in X, y \in Y}$$

stochastischer $(|S|, |S|)$ -Matrizen

$$W(x, y) = (W_{s, s'}^{x, y})_{s, s' \in S} \times S \text{ zusammen.}$$

An die Stelle des Anfangszustands bei endlichen Automaten tritt hier eine *Zustandsanfangsverteilung* σ , wobei

$V(S) := \{\sigma: S \rightarrow \{0, 1\} \mid \sum \sigma(s_i) = 1\}$ die Menge aller Zustandsverteilungen ist.

Ein stochastisches (sequentielles) System ist das Tupel (→ *n-Tupel*) (S, X, Y, W)

Ein endlicher stochastischer Automat wird beschrieben durch das δ -Tupel

$$A = (S, X, Y, W, \sigma, F)$$

mit $\sigma \in V(S)$ und der Menge der Endzustände $F \subseteq S$.

Mit der charakteristischen Funktion $ch_F: S \rightarrow \{0, 1\}$ wird F als Spaltenvektor $C(F)$ erfaßt, um das Transformationsverhalten eines stochastischen Automaten durch die „längentreue“ Abbildung zu erhalten

$$verh_A: X^* \times Y^* \rightarrow \{0, 1\}$$

mit der Eigenschaft:

$$(u, y) \rightarrow verh_A(u, v) := \sigma \cdot W(u, v) \cdot c(F)$$

$$Lg(u) = Lg(v) = 1$$

$$\text{wobei } W(u, v) = W(u_1, v_1) \dots W(u_k, v_k)$$

das der Wortbildung $u = u_1 \dots u_k, v = v_1 \dots v_k$ entsprechende Matrizenprodukt bedeutet. Böhling

Automat, unvollständiger → Automat, endlicher

Automat, vollständiger → Automat, endlicher

Automat, zellularer

Teilgebiet: Automatentheorie und formale Sprachen

tessalation automaton

Eine reguläre Anordnung identischer *endlicher Automaten*, die alle Eingabeinformationen von einer endlichen Anzahl benachbarter Automaten erhält, bezeichnet man als zellularen Automaten. Die Nachbarschaftsverbinding der Automaten sei einheitlich in der gesamten Anordnung. Zur Beschreibung der Nachbarschaftsbeziehung kann man z.B. gruppentheoretische Eigenschaften benutzen (Interpretation der regulären (→ Menge, reguläre) Anordnung als n -dimensionales *Koordinatensystem*, *Zylinder* oder *Toroid*; i.f. als *homofomer Graph* bezeichnet). Als identische Maschinen kann man z.B. *MOORE-Automaten* benutzen, dabei Eingabemenge, Ausgabemenge und Zustandsmenge identifizieren. Ein endlich erzeugter abelscher zellularer Automat ist dann ein 5 -Tupel (→ *n-Tupel*)

$$Z = (S, G^o, *, \delta_Z, s_0)$$

mit der Zustandsmenge S ,

$G^o = \{g^1, \dots, g^m\}$ als erzeugende Teilmenge einer abelschen Gruppe G mit $*$ als Gruppenoperation zur Beschreibung der Nachbarschaft N eines Knotens g in *homoformen Graphen* durch

$$N(g) = \{g, g * g^1, \dots, g * g^m\}$$

Die lokale Überföhrungsfunktion der Zustände unter Nachbarschaftsbeziehung sei

$$\delta_Z: S^{m+1} \rightarrow S,$$

$$\text{mit } s'(g) = \delta_Z(s(g), s(g * g^1), \dots, s(g * g^m)),$$

wenn $s(g)$ der Zustand des Automaten im Knoten g ist;

$s_0 \in S$ ist der durch $\delta_Z(s_0, \dots, s_0) = s_0$ definierte Ruhezustand. Böhling

Automat, zusammenhängender → Automat, endlicher

Automated Reasoning Tool

Teilgebiet: Wissensverarbeitung, Expertensysteme

Abkürzung: ART

Entwickler: Inference Corp.

Implementierungssprachen: C, LISP: Common LISP, VAXLISP, ZetaLISP

Wissensdarstellung: frames, Objekt-Attribut-Wert-Tripel, Produktions-/Inferenzregeln, Mehrfach-Objekte in Relationen.

Inferenzen: Vererbung, Mehrfachvererbung, Hypothesen, mehrere Kontexte, Zeitlogik, Truth Maintenance, non-monotonic Reasoning.

Ablaufsteuerung: Blackboard, Aktionslisten, aktive Werte, Vor-/Rückwärtsverkettung.

Unvollständiges/vages Wissen: default Reasoning, belief rules, Sicherheitsfaktoren.

Programmierungsumgebung: Maus, Menüs, Fenster. LISP environment, different help (→ Hilfebe-fehl) levels, trace (auch grafisch), break (→ Ver-bindungsabbruch); Grafik, active images, Browser.

Anwendungen: industrielle Forschung, Echtzeit-anwendungen (→ Echtzeitverarbeitung), Militär, Ortung von Spaceshuttles. Altenkrüger

Automaten, Äquivalenz endlicher

Teilgebiet: Automatentheorie und formale Sprachen

Endliche Automaten sind äquivalent bzgl. ihres Ein-/Ausgabeverhaltens g.d.w. sie für jedes Eingabewort das gleiche Ausgabewort produzieren bzw. bei Akzeptoren (→ Akzeptor, endlicher), wenn sie dieselbe Wortmenge erkennen.

Zustände eines (bzw. verschiedener) Automaten sind äquivalent, wenn von ihnen ausgehend jedes Eingabewort das gleiche Ausgabewort liefert bzw. bei Akzeptoren die Mengen der von diesen Zuständen zu Endzuständen führenden Worte gleich sind.

Automaten mit äquivalenten Zuständen können reduziert werden. Böhling

Automaten, Reduktion endlicher

Teilgebiet: Automatentheorie und formale Sprachen

Ein endlicher Automat heißt reduziert, g.d.w. es keine zwei Zustände gibt, die zueinander äquivalent sind.

Ein reduzierter endlicher Automat ist in der Klasse der deterministischen vollständigen endlichen Automaten bis auf Isomorphie eindeutig bestimmt und wird minimaler Automat genannt, da er die kleinste mögliche Anzahl von Zuständen von allen zu ihm äquivalenten Automaten besitzt. Böhling

Automatentheorie

Teilgebiet: Automatentheorie und formale Sprachen

Theorie der Automaten, vornehmlich der endlichen Automaten. Böhling

Automation

Teilgebiet: Informatik in der Betriebswirtschaft

Der durch die Automatisierung entstandene ökonomische und soziale Zustand fortgeschrittener Mechanisierung, bei dem zunehmend menschliche Tätigkeiten durch die Funktionen technischer Systeme ergänzt bzw. ersetzt werden. Der Begriff wird – 1936 entstanden – T.S. Harder, dem damaligen Produktionsleiter von General Motors (USA), zugeschrieben. Ursprünglich auf den selbsttätig erfolgenden Transport von Werkstücken zwischen Maschinen (→ Automat) angewandt, wurde der Begriff bald auf alle Fertigungsprozesse erweitert, die ganz oder teilweise unter maschineller Steuerung ohne permanente Eingriffe des Menschen ablaufen. Heute versteht man darunter auch die Verarbeitung immaterieller Objekte (Informationen) durch ADV-Anlagen (bzw. EDV-Anlagen bzw. Computer), die wesentlicher Teil der Rationalisierung sowohl im Produktions- als auch im Büro- und Verwaltungsbereich ist. Die Automation wird oft auch als „zweite industrielle Revolution“ bezeichnet. Ihre ökonomische Bedeutung besteht in der Steigerung der Arbeitsproduktivität, der damit möglichen betrieblichen Wirtschaftlichkeitsverbesserung (→ Wirtschaftlichkeitsanalyse) und der gesamtwirtschaftlichen positiven Wachstumseffekte. Ihre sozialen Auswirkungen liegen vor allem in der Veränderung der Beschäftigungsstruktur (Änderung der Qualifikationsstruktur, Arbeitslosigkeit, Arbeitszeitverkürzung). Die Automation ermöglichte nicht nur den rationelleren Ablauf vorgefundener Prozesse, sondern auch die Erfüllung von Aufgaben, die wegen ihrer Komplexität und/oder ihres Umfangs mit anderen Mitteln nicht realisierbar wären (z.B. Massendatenverarbeitung, Raumfahrt, moderne Waffensysteme). BIFOA

Automation, juristische

Teilgebiet: Informatik in Recht und öffentlicher Verwaltung

legal automation

Synonyme: Automation juristischer Entscheidung; Rechtsautomation

Siehe auch: Verwaltungsautomation

Automatisierung (Automationunterstützung) juristischer Entscheidungsprozesse.

Erläuterung:

Prozeß der fortschreitenden Ersetzung menschlicher Tätigkeit im juristischen Bereich, d.h. den Bereichen der Legislative (Normsetzung, Planung, Entscheidung, vor allem in den Parlamen-

ten), Exekutive (Verwaltung: Normanwendung (→ *Normanwendung, automationsunterstützte*)) und Judikative (Normanwendung durch Gerichte). Juristische Automation ist also nicht auf Justizautomation beschränkt. Im übrigen umfaßt sie sowohl die Automation der juristischen (End-) Entscheidung als auch die Automation ihrer Vorbereitung durch Sammlung und Bereitstellung von entscheidungsrelevantem Material bzw. Entscheidungshilfen (*Dokumentation*).

Kontext:

Die Situation im rechtlichen Bereich der Gesellschaft ist u.a. gekennzeichnet durch Informationskrise und Informationslawine (d.h. zu wenig relevante und/oder zu viel irrelevante Information). Um dieser Probleme Herr zu werden und die Aufgaben von Rechtssetzung, Rechtsanwendung und Rechtsprechung in rechtsstaatlicher Weise lösen zu können, wird die juristische Automation angestrebt, von der man eine wesentliche Rationalisierung, in gewissen Grenzen auch eine Optimierung des juristischen Verfahrens und der daraus resultierenden juristischen Entscheidungen erwartete, wovon aber auch negative Auswirkungen ausgehen (z.B. Bürokratisierung, Schablonisierung, Dequalifizierung und Arbeitslosigkeit auch im Justizbereich). Doch ist wie häufig im staatlichen Bereich Geldmangel der beste Verfassungsschutz, außer im *Geheimbereich*.

Arten:

- Justizautomation: Automation auf dem Gebiet der Justiz, d.h. der Judikative; Möglichkeiten vor allem durch *juristische Dokumentation* und juristische Fakteninformation. Die vollständige Automation der rechtlichen Entscheidungsprozesse ist derzeit nicht lösbar.
- Verwaltungsautomation: Automation auf dem Gebiet der (öffentlichen) Verwaltung. Parlamentsautomation = Automation im Bereich der Legislative: siehe Normsetzung, automationsunterstützte. Steinmüller

Automation juristischer Entscheidung → *Automation, juristische*

Automationsgrad

Teilgebiet: Informatik in der Betriebswirtschaft
Der Anteil automatisierter (→ *Automatisierung*) Tätigkeiten an der Gesamtheit aller in einem betrieblichen Teilbereich, einem Unternehmen, einer Branche oder einem Wirtschaftssektor ausgeübten Tätigkeiten ökonomischer *Leistungserstellung*. Wird beispielsweise durch die Einführung einer automatisierten Anlage die bisher von 80 Menschen erfüllte Teilaufgabe eines Produktionsprozesses nun durch die Maschine und 60 Ar-

beitskräfte erfüllt, so ist ein Automationsgrad von 25% für die entsprechende Fertigungsabteilung erreicht. Da *Automation* stets nur das relativ selbsttätige Ausführen von Funktionen durch Maschinen (→ *Automat*) kennzeichnet, ein planmäßiges oder ereignisabhängiges Eingreifen des Menschen also nicht ausschließt, und da außerdem eine Vielzahl von Tätigkeiten aus technischen oder wirtschaftlichen Gründen nicht automatisierbar sind (und für die absehbare Zukunft nicht sein werden), ist ein Automationsgrad von 100% nur als theoretischer Grenzwert anzusehen. BIFOA

automationsunterstützte Gesetzgebung

→ *Normsetzung, automationsunterstützte*

automationsunterstützte Normanwendung

→ *Normanwendung, automationsunterstützte*

automationsunterstützte Normsetzung

→ *Normsetzung, automationsunterstützte*

automationsunterstütztes Informationssystem

→ *Informationssystem, rechnergestütztes*

automationsunterstützte Subsumtion

→ *Normanwendung, automationsunterstützte*

automatische Indexierung

→ *Indexierung*

automatische Klassifikation

→ *Klassifikation, automatische*

automatische Programmierung

→ *Programmierung, automatische*

automatische Prozeßsteuerung

→ *Prozeßsteuerung, automatische*

automatische Silbentrennung

→ *Silbentrennung, automatische*

automatischer Empfang

→ *Empfang, automatischer*

automatisierte Normanwendung

→ *Normanwendung, automationsunterstützte*

Automatisierung

Teilgebiet: Informatik in der Betriebswirtschaft
automation

Der Prozeß zunehmender Ergänzung bzw. Ersetzung menschlicher Tätigkeiten durch maschinelle Funktionen. Automatisierung kennzeichnet den realtechnischen Vorgang, der zur *Automation* führt. Dabei werden Arbeits- und Produktionsprozesse so gestaltet, daß der Mensch weder permanent noch zu genau festlegbaren Zeitpunkten in den Funktionsablauf technischer Systeme einzugreifen braucht. Durch die Funktionsteilung zwischen Mensch und Maschine wird eine große-

re Schnelligkeit, Sicherheit und Wirtschaftlichkeit der Prozesse sowie eine Entlastung des Menschen von Routinearbeiten angestrebt. Die Automatisierung ist damit die konsequente Fortsetzung eines Entwicklungsprozesses der Technik, der mit der Ersetzung körperlicher Arbeit des Menschen durch Kraftmaschinen begann. War es noch das Ziel der Mechanisierung, dem Menschen die Aufgabenerfüllung mit Hilfe technischer Mittel zu ermöglichen, so beinhaltet die Automatisierung das umfassendere Ziel einer weitgehend selbsttätigen Aufgabenerfüllung durch technische Mittel. Die wichtigsten Anwendungsgebiete sind neben der industriellen Fertigungstechnik die Förder- und Verkehrstechnik sowie die Informationstechnik. Insbesondere die Erfassung (→ *Datenerfassung*), *Verarbeitung* und Übermittlung von Daten bzw. Informationen in Form von Texten, Grafiken und *Bewegtbildern* wird zunehmend automatisiert, wobei nicht nur die Steuerung industrieller Prozesse, sondern vor allem auch die Anwendung automatisierter Informations- und Kommunikationstechniken für administrative Prozesse mehr und mehr an Bedeutung gewinnt. BIFOA

Automatisierungsanweisung

Teilgebiet: Prozeßdatenverarbeitung

Anwenderorientierte Sprachelemente, die dem Anwender zur *Realisierung* seiner Aufgabe unmittelbar zur Verfügung stehen (angelehnt an VDI 2880). Seifert

Automatisierungsgerät → Prozeßrechner

Automatisierungsgerät, speicherprogrammierbares

Teilgebiet: Prozeßdatenverarbeitung
programmable controller

Synonym: Steuerung, speicherprogrammierbare
Ein elektrisches *Betriebsmittel*, das überwiegend für *Automatisierungsaufgaben* eingesetzt wird und das mittels einer anwenderorientierten *Programmiersprache (Fachsprache)* gem. seiner jeweiligen *Automatisierungsaufgabe* programmiert werden kann.

Das Automatisierungsgerät (→ *Prozeßrechner*) umfaßt auch die Versorgung(en), die Prozeßanpassung(en) und – soweit erforderlich – Programmier- und Testeinrichtungen (→ *Testen*).
Eine *anwenderorientierte* Programmiersprache orientiert sich an Sprachelementen, die bei Anwendern üblicherweise für Aufgabenbeschreibungen Verwendung finden (Automatisierungsanweisungen) (angelehnt an VDI 2880). Seifert

autonomer Automat → Automat, endlicher

Autorensprache

Teilgebiet: Computergestützter Unterricht und Pädagogik

author language

Die Autoren von Lehrstrategien (→ *Lehrprogramme, Klassifikation von*) des Computerunterstützten Unterrichtes sollen in der Lage sein, diese ohne spezielle Kenntnis einer Computersprache zu formulieren. Bei tutoriellen Lehrstrategien geschieht dies mit Hilfe einer Autorensprache: der Autor formuliert hier (in einem festen Schema) die an den Adressaten zu stellende Frage; er gibt dann eine Anzahl korrekter, falscher und möglicher Antworten vor und formuliert, welche Kommentare dem Adressaten ausgegeben werden, wenn er eine der vorgegebenen Antworten gibt. Auch Verzweigungen der didaktischen Strategie können durch die Autorensprache organisiert werden. Oftmals kann eine *Statistik* der Adressatenantworten programmiert werden; die Verzweigungen der Lehrstrategie können dann auf Grund einer statistischen Antwortauswertung gesteuert werden. Zu einer Autorensprache gehört in der Regel auch ein Tischrechner-Modus, der dem Adressaten wie dem Autor zur Verfügung steht. Die Formulierungen in der Autorensprache werden vom Rechner bei der Darbietung der Lehrstrategie interpretativ (→ *interpretative Ausführung*) abgearbeitet oder zuvor in eine maschinennahe (→ *Maschinenabhängigkeit*) Sprache übersetzt (→ *Übersetzer*). Beispiele für Autorensprachen: *COURSEWRITER III, PLANIT, COPI, TUTOR, LIDIA*. Gunzenhäuser

Autorensystem

Teilgebiet: Computergestützter Unterricht und Pädagogik

authoring system

Ein Autorensystem ist ein spezielles Anwenderprogramm (→ *Anwendungsprogramm*), das die Erstellung von Lehrprogrammen (→ *Lehrprogramme, Klassifikation von*) unterstützt. Mittels einfacher *Kommandos* und Bildschirmformulare kann ein Autor damit einzelne *Lehreinheiten* und auch ganze Lehrprogramme erzeugen.

Für die Lehreinheiten stehen meist mehrere *Formulartypen* zur Verfügung, je nachdem, ob nur Text ausgegeben werden soll, ob die zugehörige Frage eine Ja-/Nein-, eine Auswahl- oder eine frei formulierte Antwort erwartet. Bei der frei formulierten Antwort führt der Autor verschiedene *Schlüsselwörter* auf, anhand derer das Lehrprogramm die Antwort überprüft.

Oft delegiert der Autor das Sammeln von Lernerdaten an das Autorensystem.

Der Einsatz eines Autorensystems zur Erzeugung

Autorestart

eines Lehrprogramms beschränkt den Autor in der Regel auf eine vorgegebene Lehrstrategie, sein Lern- und Arbeitsaufwand wird jedoch erheblich vermindert.

Gunzenhäuser

Autorestart

Teilgebiet: Grundlagen von Informationssystemen

Zu Beginn jeder Datenbanksitzung werden automatisch alle in der vorhergehenden Datenbanksitzung unvollständig durchgeführten Datenbankbefehle (z.B. durch Systemabbruch bedingt) zurückgenommen, um die Datenbanksitzung an einem definierten Kontrollpunkt fortzuführen.

Freiburg

Autotypie

Teilgebiet: Textverarbeitung und Büroautomation

half-tone reproduction

Synonym: Rasterätzung

Um *Halbton*vorlagen (Fotografien) im Hochdruck wiedergeben zu können, werden die Schwarz- und *Grauwerte* mit Hilfe eines *Rasters* in unterschiedlich große Punkte zerlegt. Die nicht druckenden Zwischenräume werden in der Druckform chemisch vertieft.

Ehlers

Autovektor

Teilgebiet: Betriebssysteme
autovector

Vektor-Interrupt, bei dem das *Lesen* der Vektornummer wegfällt und der diesem Interrupt (\rightarrow *Unterbrechung*) zugeordnete *Befehlszähler* aus der Vektortabelle direkt in das *Steuerwerk* des Rechners (\rightarrow *Datenverarbeitungsanlage, digitale*) geladen wird.

Burkhardt

Avalanched Induced Migration

Teilgebiet: Rechner-technologie
Abkürzung: AIM

Avalanched Induced Migration bedeutet soviel wie „Wanderung von Daten durch Materieteilchen“, hervorgerufen durch den Lawinendurchbruch. AIM bezeichnet ein *Verfahren* zur Programmierung von bestimmten bipolaren (\rightarrow *bipolare Technologie*) *PROMs*. Eine *Speicherzelle* solcher *PROMs* besteht aus einem NPN-Transistor, dessen Basis nicht angeschlossen ist, während der *Kollektor* mit der Zeilenleitung und der

Emitter mit der Spaltenleitung verbunden ist. Bei der Programmierung wird das Emitterpotential so stark angehoben, daß die Basis-Emitter-Strecke in den sogenannten zweiten Durchbruch gebracht wird und Aluminiumatome von der Emitterkontaktierung zur Basis wandern. Die noch vorhandene Basis-Kollektor-Diode stellt dann das *unidirektionale* Koppellement zwischen Zeilen- und Spaltenleitung dar.

Tröller

AV-Medium \rightarrow *Audio-visuelles Medium*

Axiom \rightarrow *Axiomensystem*

axiomatische Semantik \rightarrow *Semantik, axiomatische*

Axiomenschema \rightarrow *Semantik, axiomatische*

Axiomensystem

Teilgebiet: Logik
axiom system

Ein Axiomensystem ist eine Menge von Formeln, die eine Klasse von Strukturen durch Grundeigenschaften beschreibt (z.B. wird die Klasse aller Gruppen durch die Gruppenaxiome beschrieben).

Logische Axiome sind *allgemeingültige Formeln*, die beim Aufbau eines *Logikkalküls* Verwendung finden.

Beispiel: $\neg A \vee A$ als Axiom der *Aussagenlogik*.

Müller

Axonometrie

Teilgebiet: Computer Graphics, CAD und Bildverarbeitung

Eine rechtwinklige parallele (\rightarrow *parallel*²) Projektion, bei der die Normale der Projektionsebene nicht parallel zu einer der Hauptachsen des zu projizierenden Objektes ist. Axonometrien können isometrisch (die Projektionsebene bildet mit jeder der drei Hauptachsen den gleichen Winkel), dimetrisch (die Projektionsebene bildet mit zwei Hauptachsen den gleichen Winkel) oder trimetrisch (die Projektionsebene bildet mit jeder der Hauptachsen einen anderen Winkel) sein.

Gorny

B&B → *Branch and Bound-Methode*

B*-Baum

Teilgebiet: Grundlagen von Informationssystemen

B-tree*

Siehe auch: B-Baum

Ein B*-Baum ist eine Modifikation des B-Baums, die eine höhere Verzweigung und damit einen flacheren Baum und ein verbessertes Zugriffsverhalten zum Ziel hat. Dies wird erreicht, indem man nur in den Blattknoten (→ *Blatt eines Baumes*) die vollständigen Einträge aufführt. In den Nichtblattknoten (Indexteil (→ *Schlüssel*¹) des B-Baums) werden nur die Schlüssel der Einträge abgelegt. Der dadurch entstehende Platzgewinn läßt im Indexteil mehr Verweise (→ *Verweis*²) pro Knoten zu als beim reinen B-Baum.

Munz

BaB → *Bausteinebank*

BaBMS → *Bausteinebank-Management-System*

Babylon

Teilgebiet: Wissensverarbeitung, Expertensysteme

Entwickler: GMD

Implementierungssprache: ZetaLISP

Wissensdarstellung: frames, Produktionsregeln, Relationen, constraints

Ablaufsteuerung: aktive Werte, Vor-/Rückwärtsverkettung

Unvollständiges Wissen: defaults

Programmierungsumgebung: Maus, Menüs, vordefinierte Fragen des Systems, kurze Benutzerantworten ohne vielfältige natürlichsprachliche Syntax.

Altenkrüger

Backtracking

Teilgebiete: Kognitive Methoden – Künstliche Intelligenz; Linguistische Datenverarbeitung; Wissensverarbeitung, Expertensysteme
Siehe auch: Zurücksetzung; Rücksetzen

Zurückgehen in einem Lösungsbaum an einen bestimmten Knoten, von dem aus erneut eine Lösung gesucht wird. Backtracking ist von grundlegender Bedeutung bei allen Problemlöseverfahren mit (*heuristischer*) Suche, so z.B. auch in PROLOG. Dependency Directed Backtracking versucht Wissen einzusetzen, um einen möglichst geeigneten Rückkehrpunkt zu finden.

Hellwig; Lenz, A.; Richter, M.; Schmitz

backtracking point → *Verzweigungspunkt*

Backup → *Sicherstellung*

Back-up-Rechenzentrum → *Ausfall-Rechenzentrum*

Backup-System → *Sicherstellung*

Backus-Naur-Form

Teilgebiet: Programmierung

Backus-Naur-form

Synonym: BNF-Beschreibung

Abkürzung: BNF

Notationsform für *Ersetzungsregeln* kontextfreier Grammatiken zur Definition der Syntax von Programmiersprachen.

Die Backus-Naur Form wurde im Rahmen der Definition der Programmiersprache ALGOL-60 als *Metasprache* zur formalen Festlegung der Syntax der Programmiersprache entwickelt.

In der Metasprache werden Produktionsregeln definiert, aus denen syntaktisch korrekte Zeichenreihen (→ *Zeichenkette*) (Wörter) der zu definierenden Sprache abgeleitet werden können. Sie entspricht in ihrem Aufbau den *kontextfreien* Chomsky-Grammatiken.

Jede *Produktionsregel* definiert ein auf ihrer linken Seite stehendes Nichtterminalzeichen (siehe *nichtterminales Zeichen*) durch eine auf der rechten Seite stehende Zeichenkette, rechte und linke Seite werden durch das Definitionszeichen „::=" getrennt.

Nichtterminalzeichen (auch als metasprachliche Variablen oder syntaktische Objekte bezeichnet) werden in spitze Klammern "<“, „>“ eingeschlossen. Alternativ gültige rechte Seiten werden durch einen senkrechten Strich „|“ voneinander getrennt. Die Zeichen >, <, |, ::= gehören nicht zum Alphabet der zu definierenden Sprache.

Beispiel:

Die nachfolgenden Zeilen definieren metasprachlich das syntaktische Objekt *Bezeichner* (identifizier) in der Programmiersprache PASCAL

(1) <ziffer> ::= 0|1|2|3|4|5|6|7|8|9

(2) <buchstabe> ::= a|b|c|d|...|x|y|z

(3) <wort> ::=
<buchstabe>|<ziffer>|
<wort><buchstabe>|
<wort><Ziffer>

(4) <bezeichner> ::= <buchstabe>|<buchstabe><wort>

Die Produktionsregel (1) besagt, daß das syntaktische Objekt <ziffer> entweder als Alphabetzeichen 0 oder 1 oder ... oder 9 ist. Die Regel (2)

legt fest, was ein syntaktisches Objekt $\langle \text{buchstabe} \rangle$ ist. Da die Elemente der rechten Seiten von (1) und (2) nicht in spitzen Klammern stehen, handelt es sich um Terminalzeichen. Das sind jene Zeichen, aus denen die Zeichenfolgen der zu definierenden Sprache gebildet werden. Die Produktionsregel (3) gibt an, wie das syntaktische Objekt $\langle \text{wort} \rangle$ gebildet wird. Dies ist ein Nichtterminalzeichen, das zur Bildung der Produktionsregel (4) benutzt wird. Um gültige Bezeichner der Programmiersprache abzuleiten, werden in den möglichen rechten Seiten von (4) Nichtterminalzeichen durch zulässige Definitionen der Regeln (1), (2) oder (3) *substituiert*.

Beispiele zur Ableitung:

$\langle \text{bezeichner} \rangle \rightarrow \langle \text{buchstabe} \rangle \rightarrow a$

(a ist ein zulässiger Bezeichner)

$\langle \text{bezeichner} \rangle \rightarrow \langle \text{buchstabe} \rangle \langle \text{wort} \rangle \rightarrow x$

$\langle \text{wort} \rangle \rightarrow x \langle \text{wort} \rangle \langle \text{ziffer} \rangle \rightarrow x \langle \text{wort} \rangle 7 \rightarrow x$

$\langle \text{buchstabe} \rangle 7 \rightarrow xy7$

(xy7 ist ein zulässiger Bezeichner)

Informal ist ein Bezeichner eine beliebige Folge von *Buchstaben* und *Ziffern*, die stets mit einem Buchstaben beginnt.

Im Verlaufe der Entwicklung wurden die Ausdrucksmittel der metasprachlichen *Darstellung* erweitert (erweiterte Backus-Naur-Form (EBNF)):

- (1) Zeichenfolgen, die optional sind, werden in eckige Klammern eingeschlossen
- (2) Zeichenfolgen, die beliebig oft (auch nullmal) wiederholt werden können, werden in geschweifte Klammern eingeschlossen.

Eckige und geschweifte Klammern dürfen ebenfalls nicht zum Alphabet der zu definierenden Sprache gehören.

Mit diesen Erweiterungen kann die Produktionsregel (4) wie folgt angegeben werden:

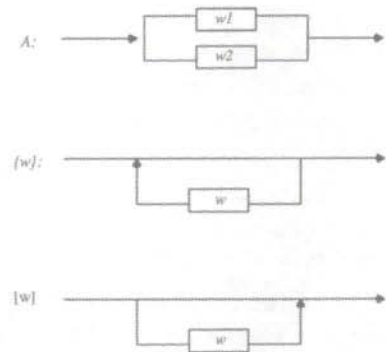
- (5) $\langle \text{buchstabe} \rangle ::= \langle \text{buchstabe} \rangle \{ \langle \text{buchstabe} \rangle | \langle \text{ziffer} \rangle \}$

Der Syntax in der EBNF-Notation kann in eine *graphische Darstellung* umgesetzt werden. Dafür wird ein *gerichteter Graph* mit zwei Knotentypen benutzt. Terminalzeichenfolgen werden in „runde“ *Knoten* und Nichtterminale in „viereckige“ *Knoten* eingetragen. Die Aufeinanderfolge (*Verkettung*) von Zeichenfolgen wird durch „ \rightarrow “ markiert.

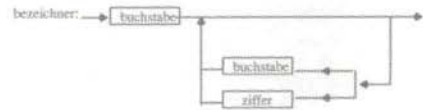
Dann gelten folgende Regeln:

$\langle A \rangle ::= w1 | w2$

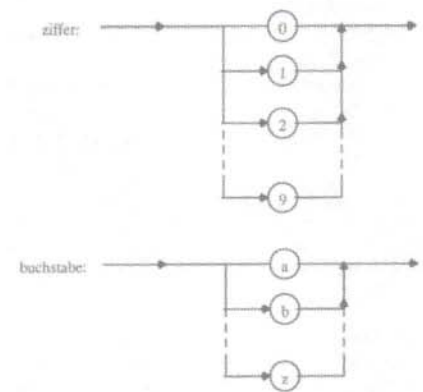
$\langle A \rangle ::= w1 | w2$



Die Produktionsregel (5) in EBNF-Notation wird durch folgenden *Graph* beschrieben:



Und für die Regeln (1) und (2) gilt:



Anmerkung: Anschaulich können die Regeln von BNFs durch Syntaxdiagramme dargestellt werden, wobei ein Wort der erzeugten Sprache einem vollständigen Durchlauf des dem Startzeichen zugeordneten Diagramms entspricht. Diese Syntax-Graphen haben den Vorteil, daß keine *Kollisionen* zwischen Terminal- und Nichtterminalzeichen auftreten und syntaktische richtige Objekte der zu definierenden Sprache dadurch erzeugt werden, daß man mögliche Wege durch den Graphen beschreitet.

Die BNF kann als rekursives Gleichungssystem für die den Nichtterminalen entsprechenden Wortmengenvariablen aufgefaßt werden. Die generierte Sprache entspricht dann der Lösung des Gleichungssystems für das nichtterminale Startzeichen. Natürlich gibt es verschiedene Möglichkeiten, ein und dieselbe Sprache durch BNFs zu beschreiben.

Durch BNFs kann die Syntax von Programmiersprachen kontextfrei beschrieben werden, wodurch formale Methoden zur Korrektheitsprüfung (→ *Korrektheitsbeweis*) und *Semantik* definition anwendbar werden. Die *Korrektheit* eines Programms in einer Sprache, die durch eine BNF vorgegeben wird, kann dann durch einen *Kellerautomaten* überprüft werden, der wiederum aus der BNF konstruierbar ist. Gerber; Mätzl

Backus-Normal-Form → *Backus-Naur-Form*

Backus-Normalform → *Backus-Naur-Form*

Backward Chaining

Teilgebiete: Kognitive Methoden – Künstliche Intelligenz; Wissensverarbeitung, Expertensysteme

Siehe auch: Rückwärtsverkettung

Backward Chaining dient bei *computergestützten Problemlösungsprozessen* der *Analyse* des Lösungswegs. Ausgehend von dem erzielten Ergebnis wird durch *Schlußfolgerung* der Lösungsweg zurückverfolgt und die lösungsrelevanten Daten identifiziert. Freiburg

backward recovery → *roll back*

BACON

Teilgebiet: Informatik in Recht und öffentlicher Verwaltung

Back FILE Conversion Project

Abkürzung von: BACON-Projekt

Projekt der 1986 vereinbarten trilateralen Zusammenarbeit des Europäischen Patentamts (EPA), des United State Patent and Trademark Office (USPTO) und des Japanischen Patentamts (JPO) auf dem Gebiet der *Patentinformation*. Gegenstand ist die Erfassung (→ *Datenerfassung*) von seit 1920 weltweit veröffentlichten Patentdokumenten in einem *Archivsystem* im Faksimile-Format. Ziel ist die Speicherung von 5,9 Mio. Patentdokumenten verschiedener europäischer Länder (Frankreichs, Deutschlands, der Schweiz, Großbritanniens), des Europäischen Patentamts und der *WIPO*, und die Erfassung von 3,2 Mio. US-amerikanischer und 3,3 Mio. japanischer Patentdokumente mit einem Gesamtvolumen von rund 60 Mio. Seiten (Stand 1989). Parallel (→ *parallel*²) dazu werden nun auch die Neuein-

gänge (Frontfile) digitalisiert, um eine vollständige Faksimile-Datenbank zu erhalten.

Ein im Rahmen des BACON-Projekts vereinbarter Standard zur Weitergabe zeichencodierter Texte (insbesondere von Patentanmeldungen auf *Datenträgern*) trägt die Kurzbezeichnung *Datimtex* (Data images text). Auf der Grundlage von *WIPO*-Standards und weiterer internationaler Normen wird der Volltext um (druckbare) *Steuerzeichen* – z.B., Zeichen für das Druckbild, für eingefügte Formeln und Zeichnungen und zur Markierung (→ *Markieren*) bestimmter Teile (bspw. der Ansprüche) – erweitert, um die Wiedergabe eines *Dokuments* durch Zusammenführen von Text- und Bilddaten zu ermöglichen. Häußer

BACON-Projekt → *BACON*

Bahnsteuerung

Teilgebiet: Prozeßdatenverarbeitung

Steuerung eines Objekts derart, daß es eine a priori vorgegebene Bahn durchläuft.

Bahnen in *Zustandsräumen* werden als *Trajektorien* bezeichnet. Speziell: Bahnsteuerung bei Werkzeugmaschinensteuerung im Gegensatz zu Punkt- bzw. Streckensteuerung. Lauber

BAKKALAUREUS

Teilgebiet: Computergestützter Unterricht und Pädagogik

Für den Schuleinsatz entwickelte *Nixdorf* in den 60er Jahren einen Lehrautomaten *BAKKALAU-REUS*, welcher vergleichsweise einfache Unterrichtsprogramme (→ *Unterricht, computerunterstützter*), präsentiert über einen Bildprojektor und ein Tonbandgerät, ausführen konnte. An das System (eine Zentraleinheit 820 mit Platten (→ *Magnetplattenspeicher*) und *Drucker*) können bis zu 64 Adressatenplätze angeschlossen werden, wobei jeder Platz über sieben Eingabetasten (+/-ABCD, wobei A als Ja, D als Nein bezeichnet ist) verfügt, über die individuelle Eingaben (im Sinne einer *Auswahlantwort*) zum jeweiligen Stand des Lehrprogramms (→ *Lehrprogramme, Klassifikation von*) möglich sind.

Das System *BAKKALAU-REUS* wurde Anfang der 70er Jahre in einigen Schulen erprobt, ist jedoch heute nicht mehr im Einsatz. Schneider

Balkencode

Teilgebiet: Allgemeines

barcode

Synonym: Strichcode

Normierte *Schriftzeichen* eines maschinell-optisch erkennbaren Lesesymbols, das richtungs- und lageunabhängig gelesen werden kann. Jedes

Balkendiagramm

Zeichen besteht aus einer Gruppe von Balken und Zwischenräumen. *Ehlers*

Balkendiagramm → *Präsentationsgraphik*

Balloting-Phase → *ISO*

Ballungsanalyse → *Musteranalyse*

Bananenprinzip

Teilgebiet: Programmierung

Dem Bananenprinzip liegt die Idee zugrunde, Standard-Software bei dem *Anwender* reifen zu lassen. Dabei wird *Software* vor der Freigabe nur oberflächlich *getestet*, oder es wird nicht die gesamte erforderliche *Leistung* angeboten, oder es wird nur gering auf *Benutzerfreundlichkeit* geachtet. Nach Verbesserungswünschen oder Beschwerden wird dann eine verbesserte *Version* der Software auf den Markt gebracht, die von bestehenden Anwendern gegen eine Zuzahlung erworben werden kann.

Bei manchen *Produkten* wurde diese Methode über einen längeren *Zeitraum* angewendet, so daß die Kunden schließlich sogar ein Mehrfaches des ursprünglichen Preises zu zahlen hatten.

Hausherr

Band

Teilgebiet: Rechnerperipherie

Der *Begriff* des Bandes wird in der Informatik bei Speichermedien für Daten und für Kontrollinformation (→ *Steuerinformation*) gebraucht, z.B. bei Daten auf Magnet-, Papier- oder Kunststoffband oder Lochstreifen, für Kontrollinformation auf Papierband zur Steuerung des Papiervorschubs bei *Schnelldruckern* und als gedachtes Steuerband für Turing-Maschinen (→ *Automat*).

Burkhardt

Bandbreite

Teilgebiet: Datenübertragung
bandwidth

Die Bandbreite eines Signals ist derjenige Frequenzbereich, in dem die gesamte oder nahezu gesamte Leistung eines Signals enthalten ist. Häufig wird die Bandbreite reduziert, um Übertragungsaufwand zu sparen. In der Nachrichtentechnik unterscheidet man zwischen einer Schmalband- und einer Breitbandkommunikation, wobei unterschiedliche Definitionen verwendet werden. Meist spricht man jedoch von Breitbandkommunikation, wenn die Bandbreite des zu übertragenden Signals (und damit auch des Übertragungskanal) im Megahertz-Bereich liegt – dies ist z.B. der Fall bei der schnellen Daten- oder *Bewegbild*-Übertragung. Dagegen stellt z.B. die *Datenübertragung* im Fernsprechkanal (Band-

breite 3,1 kHz) eine Schmalbandübertragung dar. Auch bei der digitalen Übertragung verwendet man die Begriffe „Schmalband“ und „Breitband“, obwohl die kennzeichnende Größe die *Bitrate* ist. *Fellbaum*

Bandmarke → *Automat, linear beschränkter*

Bandspreiztechniken

Teilgebiet: Datenübertragung
spread spectrum techniques

Um eine höhere Unabhängigkeit von *Störungen* auf dem Übertragungskanal zu erzielen, werden Bandspreiztechniken eingesetzt. Bei der Übertragung über große Entfernungen mit geringeren *Leistungen* im Weltraum verwendet man für die zu übertragenden *binären Zeichen* Rauschcodes großer Länge, so daß die Übertragung zwar langsam wird, aber bei geringer Leistung relativ fehlerfrei erfolgt.

Ein künftiges Anwendungsgebiet wird die Mobilkommunikation sein, bei der durch orthogonale rauschartige Codes viele *Teilnehmer* den gleichen Übertragungskanal verwenden und durch sog. Codemultiplex voneinander zu trennen sind. Neben dem hier genannten Verfahren mit rauschartigen Codes zählen zu den Bandspreiztechniken auch Frequenz- und Zeitsprungverfahren, engl. frequency- bzw. time hopping. *Kroschel*

Banking-POS

Teilgebiete: Betrieb von Rechensystemen; Datenübertragung

point of sale banking

Synonyme: Abrechnungssystem, elektronisches; Ladenkasse, bargeldlose Bezahlung an der

POS-Banking stellt ein Geldeinzugsverfahren beim bargeldlosen Zahlungsverkehr dar. Mit Hilfe von „POS-Systemen“, die die auf einem Magnetstreifen (→ *Magnetstreifenspeicher*) (oder *Chip*) enthaltenen Informationen eines Bankkarteninhabers, wie Kontonummer, Bankleitzahl und *persönliche Identifikationsnummer*, lesen können, lassen sich Rechnungsbeträge praktisch an der Ladenkasse via Datenfernübertragung (→ *Datenübertragung*) zwischen einer „Computerkasse“ und einem Kreditinstitut abbuchen.

Neben dem Vorteil eines schnellen Zahlungseinzugs (Direktbuchung) für Händler trägt das *Verfahren* auch zur Reduzierung hinsichtlich der Betrugsgefahr bei, da eine *elektronische* Bezahlung nur dann erfolgt, wenn das Kundenkonto ein Guthaben bzw. einen offenen Kreditrahmen aufweist. Die Banken sehen die Vorteile maßgeblich in der Minderung von Bargeldabhebungen ihrer Kunden sowie in der Reduzierung der Belegverarbeitung. *Adena*

Barcode → *Balkencode*

Bargeldbörse, elektronische

Teilgebiet: Informatik im Bankwesen

Der Gegenwert eines Geldbetrags wird *elektronisch* gespeichert (z.B. auf einer *Chipkarte*) und ähnlich einer Bargeldbörse anonym transferiert. *Martiny*

Baseband-Verfahren

Teilgebiet: Datenübertragung

Verfahren der *Signal-Codierung*, bei dem die *Bit* werte der übertragenen Daten nicht durch Modulationsverfahren ein „Trägersignal“ verändern, sondern das Signal nur zwischen zwei diskreten Pegelwerten steuern. Daten werden bei optischer Übertragung und allgemein im *LAN* im Baseband-Verfahren übertragen. *Nowak*

BASEX

Teilgebiet: Prozeßdatenverarbeitung

Abkürzung von: BASIC for Experiments

Eine *höhere Programmiersprache* zur Prozeßsteuerung, die auf BASIC aufbaut (BASIC for Experiments). *Lauber*

BASIC

Teilgebiet: Programmierung

Abkürzung von: beginner's all-purpose symbolic instruction code

BASIC ist eine leicht erlernbare problemorientierte *Programmiersprache*. 1965 wurde sie am Dartmouth College, Hanover N.H. (USA), als *Dialog* form von *FORTRAN* ursprünglich mit dem Ziel entwickelt, Anfängern das Programmieren zu erleichtern.

Mittlerweile dient BASIC aufgrund verschiedener Erweiterungen in den USA als Standard-Programmiersprache für dialogfähige Kleincomputer (→ *Minicomputer*), für Aufgabenstellungen aus mathematisch-technischem wie kommerziellem Bereich gleichermaßen geeignet.

Sprachelemente sind Befehle, *Konstanten* und Variablen (→ *Variable*¹).

Eine *Standardisierung* der Sprache wie z.B. bei COBOL ist noch nicht vorgesehen, doch sehen die meisten *Hersteller* neben Matrix-Operationen *Unterprogramm*technik, Standard- und Benutzerfunktionen sowie die Möglichkeit der *formatisierten Ausgabe* für die BASIC-Version ihrer DVA vor. *Schmitz*

BASIC for Experiments → *BASEX*

BASIC FORTRAN → *FORTRAN*

BASIC PEARL → *PEARL*

Basisadresse

Teilgebiete: Programmierung; Rechnerorganisation

base address; relocation address; segment address

Synonyme: Anfangsadresse; Bezugsadresse
Siehe auch: Adreßwiederholung; Segmentierung eines Programms; Seitenaustauschverfahren

In vielen Rechnern wird nicht eine *elementare Adresse* (Absolutadresse) im *Befehls*wort mitgeführt, sondern nur eine *Relativadresse* in einem an anderer Stelle bereits definierten Teilbereich des Speichers. Die Anfangsadresse dieses Teilbereichs nennt man Basisadresse oder Segmentadresse.

Die Vorteile dieser Technik sind:

1. Der *Adreßteil* des Befehlswortes kann auch bei großem realem *Adreßraum* relativ klein sein.
2. Die Verwaltung des realen bzw. *virtuellen Adreßraums* wird erleichtert, wenn für die Basisadresse (n) eigene *Register* (*Basisregister*, Segmentregister) zur Verfügung stehen und für die Adreßumsetzung (→ *Adreßbestimmung*) ein eigenes Adreßrechenwerk. Bei der Programmierung wird die Basisadresse als die Adresse eines ausgezeichneten Speicherplatzes (z.B. des ersten Befehls eines Programms, des ersten Datums (→ *Dateneinheit*) in einem Datenbereich), meist aufgenommen in ein Register, verwendet. Sie dient zusammen mit einem *Adressenversatz* der *Adressierung* irgendeines Speicherplatzes in dem jeweiligen Programm bzw. Datenbereich. In einer Adressenangabe ist (im allgemeinen) nicht die Basisadresse, sondern die Adresse des Registers, das die Basisadresse aufgenommen hat, enthalten; man erreicht dadurch eine (u.U. nicht unerhebliche) Verkleinerung von Adreßteilen im *Befehlsformat*.

Hoffmann, H.-J.; Klar

Basisadresse, offene

Teilgebiet: Rechnerorganisation

Basisadresse, die von dem sie verwendenden Prozeß verändert werden kann. *Klar*

Basisadresse, verdeckte

Teilgebiet: Rechnerorganisation

Basisadresse, die für den sie verwendenden Prozeß unzugänglich ist.

Anmerkung: Üblicherweise verwaltet das *Betriebs*system derartige Basisadressen in geeigneten *Tabellen*, wie Seitentabelle oder Segmenttabelle (→ *Segment*¹). *Klar*

Basisdaten

Teilgebiet: Informatik in der Betriebswirtschaft
Synonyme: Ausgangsdaten; Primärdaten; Urdaten

Aufgaben der Datenverarbeitung implizieren die Existenz von Basis- oder Ausgangsdaten, die über Vorgänge der *Dateneingabe* Datenverarbeitungsanlagen zugeführt werden. Diese *Definition* gilt also

1. für Daten, die zum erstmalig für einen *automatisierten* Verarbeitungsprozeß (\rightarrow *Prozeß*²) vorbereitet werden, und
2. für Daten, die Ergebnisse anderer automatisierter Verarbeitungsprozesse sind.

In jedem Fall erfordert die Zuführung von Daten zu Vorgängen der Datenverarbeitung zunächst Funktionen der *Datenerfassung* und/ oder der *Datenspeicherung*.
Schneider

Basis der Radixschreibweise \rightarrow *Radixschreibweise*

Basisflipflop \rightarrow *Taktsteuerungsarten eines Flipflops*

Basisfunktion \rightarrow *Ausgangsfunktion*

Basisinformation

Teilgebiet: Informatik in der Betriebswirtschaft
Synonym: Grunddaten

Basisinformationen sind Informationen, die in einem Prozeß mit Hilfe einer Verarbeitungsvorschrift zu aussagefähigeren Informationen verknüpft oder verdichtet (\rightarrow *Kompaktifizieren*) werden: Die Ergebnisse stellen dann häufig betriebswirtschaftliche Kennzahlen dar. So werden beispielsweise Lagerbewegungen in Lagerbestände pro Zeitperiode, Rechnungsdaten in Auszahlungssummen, Buchungen in Journalsummen überführt. Dabei können derartige Informationen für weitere Kennzahlenbildungen wieder den Charakter von Basisinformationen haben. Man kann dann von mehrstufiger *Verarbeitung* von Basisinformationen sprechen. So ist die Kennzahl (\rightarrow *Kennsatz*) ‚Lagerbestand‘ zusammen mit den ‚Lagerbewegungen‘ wieder Basisinformation für die Lagerbestandsführung.
BIFOA

Basisinterpretation \rightarrow *Petri-Netz*

Basislösung

Teilgebiet: Quantitative Methoden
basic solution

Begriff der linearen *Algebra*. Basislösung (zur Basis B) heißt jene Lösung eines linearen Gleichungssystems $A \cdot x = b$ mit m Zeilen, n Variablen (\rightarrow *Variable*¹) ($n > m$) und $\text{Rang}(A) = m$, die sich ergibt, wenn nach *Auswahl* einer nichtsingulären

$(m \times m)$ -Teilmatrix (Basis) von A die zu den restlichen $n - m$ Spalten gehörenden Variablen auf Null gesetzt werden. Die m Variablen, welche zu den Basisvektoren gehören, heißen Basisvariablen, die anderen Nichtbasisvariablen. Eine Basislösung heißt degeneriert, falls eine oder mehrere Basisvariablen Null sind.

Ein Gleichungssystem mit m Zeilen und n Variablen besitzt höchstens $\binom{n}{m}$ Basislösungen. Der Übergang von einer Basislösung zur Basis B_1 zu einer zur Basis B_2 , welche sich nur durch einen (Spalten-) Vektor von B_1 unterscheidet, heißt elementare Basistransformation oder Basistausch. Für die Umrechnung existieren verschiedene *Verfahren* (*Pivotschritt*, *Produktform der Inversen*).
Hummeltenberg

Basisregister

Teilgebiet: Programmierung
base register

Basisregister werden ähnlich den *Indexregistern* zur Modifikation von Adressen verwandt. Vor Ausführung eines Befehls wird der Inhalt des Basisregisters zu der im Befehl angegebenen *Operandenadresse* addiert und so die endgültige Adresse des *Operanden* ermittelt. Ein anderes Basisregister kann dazu verwandt werden, die gleiche Modifikation der Programmadressen durchzuführen. Basisregister erlauben die Erstellung verschieblicher Programme (\rightarrow *Swapping*). Das sind Programme, die bei ihrer Ausführung nicht an eine bestimmte Lage im *Hauptspeicher* gebunden sind. Sie werden so formuliert, als würden sie bei der Adresse 0 beginnen. Während der Ausführung wird dann zu jeder Adresse des Programms der Basisregisterinhalt addiert, der den echten Programmanfang angibt. Zum anderen sind Basisregister notwendig, um Speicher zu *adressieren*, die größer sind, als der *Adreßteil* der Befehle zuläßt. Es genügt dann, die zur Speicheradressierung erforderliche längere Adresse bei der Addition der im Programm angegebenen kürzeren Adresse mit dem hinreichend langen Basisregisterinhalt herzustellen.
Göntler

Basis-Software \rightarrow *Systemsoftware*

Batch-Accounting

Teilgebiet: Betrieb von Rechensystemen
batch-accounting

Abrechnung der in einem festen Zeitintervall durch ein oder mehrere Rechensysteme erbrachten Systemleistungen, *dargestellt* meist als Summe der Teilsystem-Leistungen. Die *Abrechnungsroutine* läuft als Batch-Routine (\rightarrow *Batch-Betrieb*).
Graef

Batch-Betrieb \rightarrow *Stapelbetrieb*

Batch-Formatierer

Teilgebiet: Textverarbeitung und Büroautomation

Als Eingabe für einen Batch-Formatierer wird eine *Datei* erstellt, in der zusätzlich zum Text Informationen integriert sind, die die Form (z.B. die *Typographie*) des Schriftstückes bestimmen. Batch-Formatierer finden vor allen Dingen bei langen Dokumenten Verwendung (*Reports*, *Manuals*). Der Batch-Formatierer interpretiert die *Datei* und stellt das endgültige *Dokument* her.

Richter, J.

Baud

Teilgebiet: Datenübertragung

baud

Abkürzung: Bd

Einheit der *Schrittgeschwindigkeit* (1 Baud = 1 Modulationsschritt/s), wobei unter Schritt (signal element (→ *Schritt*¹)) ein Signal von definierter Dauer mit einem eindeutigen *Wertebereich* von zwei oder endlich vielen vereinbarten Wertebereichen eines oder mehrerer *Signalparameter* (z.B. Amplitude, Frequenz, Phase, Polarität) verstanden wird.

Matejka

Baeinheit

Teilgebiet: Rechnerorganisation

physical unit

Ein materielles Gebilde als Betrachtungseinheit, die sich bei einer auf *Zusammensetzung* oder *Aufbau* gerichteten, also konstruktiven Betrachtungsweise ergibt. Dabei kann ein System von Baueinheiten in einem gegebenen Zusammenhang wieder als eine Baueinheit aufgefaßt werden. Der Baueinheit können eine oder mehrere Funktionseinheiten entsprechen.

Es wird empfohlen, bei Benennung bestimmter Baueinheiten in Wortzusammensetzungen vorzugsweise zu gebrauchen (in absteigender *Reihenfolge*): ...system, ...werk oder ...einrichtung, ...glied oder ...element.

Im Gegensatz zur Funktionseinheit wird die Frage, was man als Baueinheit bezeichnet, unmißverständlich beantwortet: Die Betrachtungsweise hebt den Aspekt eines materiellen Gebildes hervor. Dabei wird das Wort ‚Gebilde‘ benutzt für etwas, das aus anderem gebildet worden ist, ohne sich darüber auszulassen, aus welchen Elementen oder durch welchen Vorgang das Gebilde entstanden ist (nach *DIN 44 300 T1*).

Fleischhauer; Guse; Rouette; Wojtkowiak

Baugruppe → *Stückliste*

Baukastenprinzip

Teilgebiet: Informatik in der Betriebswirtschaft

Synonyme: Bausteineprinzip; Methodenbankprinzip

Siehe auch: Methodenbank

Im Rahmen der Gestaltung und *Nutzung* heutiger *Informationssysteme* werden die Begriffe Methodenbank, Modellbank, Bausteinebank – die Elemente des Baukastens sind die Bausteine -, Software-Bank (→ *Software-System*), Software-Informationssystem, *Software-Faktoren-Bibliothek*, Softwarearchitektur, Modul-Bank (→ *Modul*¹), *Generatorsoftware* und sogar *Chief-Programmer-Team-Organisation* teilweise überlappend und recht unterschiedlich gebraucht.

Hier soll unter Baukastenprinzip bzw. Bausteineprinzip eine spezielle Vorgehensweise zur Erstellung von individueller *Anwendungssoftware* verstanden werden: Konkrete *Anwendungsprogramme* werden z.T. durch schon existierende Teilprogramme (*Unterprogramme*), die für oft wiederkehrende Probleme im Rahmen der neu zu erstellenden Programme vorgefertigte Lösungen (Standardbausteine (→ *Standardanwendungssoftware*)) darstellen, aufgebaut. Solche Bausteine können vom *Assembler-Makro* (→ *Makroassemblierer*) über *Datenbeschreibungen* (→ *Datenbeschreibungssprache*) bis hin zur *Datenbanksoftware* (→ *Datenbanksystem*) reichen. Zu unterscheiden ist dieses Prinzip z.B. vom Prinzip der normierten Programmierung (NP (→ *Programmierung, normierte*)), bei dem durch einen *Programmgenerator* immer wieder gleiche Aufgabenfolgen bei jedem Erstellungsprozeß für ein Anwendungsprogramm generiert werden. Es gibt jedoch *Programmgeneratoren*, die neben der Unterstützung der NP und der SP (→ *Programmierung, strukturierte*) auch die Übernahme von Standardbausteinen aus einer Bausteinebank vorsehen und unterstützen. Eine besondere Problematik der Verwendung solcher Standardbausteine liegt in der Schnittstellenlösung und dem Funktionsumfang der zugrunde liegenden Standardlösung. Eine häufige Mehrfachverwendung der Bausteine (Wiederverwendung) in vielen individuellen Anwendungsprogrammen ist nur durch eine strenge Normung des Aufbaus der Bausteine – Strukturierung, z.B. durch Vorsehen von nur einem Ausgang und Eingang, den Datentransfer (→ *Datenübertragung*) nur über *Parameter* oder über *Dateien* und der Kombinierbarkeit – und einer ggf. parametergesteuerten Funktionsfestlegung gewährleistet. Eine effektive Nutzung solcher Bausteine bedarf einer guten *Dokumentation* über ihre Funktionsweise, voraussetzend des Nachweises ihrer Existenz, und der Haltung und Organisation der Bausteine in Form einer BaB (Bausteinebank) (auch:

Baukastenprinzip

Bausteinebibliothek), die über ein einheitliches BaBMS (Bausteinebank-Management-System) gewartet (→ *Wartung*) und genutzt wird. Ein solches BaBMS, verstanden als Datenpool mit Verwaltungshilfen, müßte Hilfsmittel zur *Definition*, *Manipulation*, *Ausführung*, *Auskunft*, *Zugriffskontrolle* und *Auswertung* beinhalten. Bestimmte Bibliothekssysteme leisten in dieser Richtung einiges. Die *Architektur* eines solchen BaB könnte in Analogie zur Architektur eines Datenbanksystems (DBS) aufgebaut werden (Ebenenkonzept, Einführung von Benutzerschichten etc.). Erste Ansätze gibt es bereits.

In *analoger* Weise ist die *Objektorientierung* zu sehen. Klassenkonzept und *Vererbung* realisieren die Wiederverwendung im Sinne von Bausteinen.

Die *Flexibilität* der Gestaltung (Programmierung) im Sinne einer individuellen Vorgehensweise wird durch die Nutzung von Bausteinen erheblich eingeschränkt. Jedoch erlaubt dieses Prinzip in der Umgebung einer insgesamt strukturierten Vorgehensweise die Entwicklung übersichtlicher und damit kontrollierbarer Systeme. Die entstehenden Programme werden wartungsfreundlicher, ggf. muß bei notwendigen Änderungen in mehreren Programmen nur ein Baustein (→ *Baueinheit*) geändert werden. Die *Programmerstellung* und -wartung wird im Sinne dieses multidimensionalen Zielsystems (→ *System¹*) wirtschaftlicher.

Zu unterscheiden vom Begriff der Bausteinebank ist der Begriff der Methodenbank – sie werden in der Literatur jedoch in vielen Veröffentlichungen als gleich betrachtet –: Methodenbanken, ggf. auch unter einem einheitlichen Management-System (MeBMS), stellen eine Methodensammlung zur Lösung konkreter betriebswirtschaftlicher, mathematischer etc. Fragestellungen dar. Ihre besondere Bedeutung liegt dabei in dem Angebot alternativer und sich ergänzender Auswertungsmethoden für einen Problembereich. Ein Beispiel wäre MPSX (Methodensammlung auf dem Gebiet der mathematischen Optimierung) oder BETINA (Methoden für ein Technisches Informationssystem mit Netz-Analyse). Zweifelsohne sind die Übergänge zur Bausteinebank fließend, wenn der Systemplaner als Fragesteller und das Angebot, alternative *Sortier Routinen* nutzen zu können, betrachtet wird! Im Sinne der Methodenbank versteht man unter Modellbank (MoB und MoBMS) die Zurverfügungstellung bzw. die Möglichkeit der Generierung (→ *Synthese*) von *modularen* Modellhierarchien (→ *Gegenstandsraum*), wobei die Elemente der Struktur Metho-

den sind: Einmalige und umfassende Aufgaben inhaltlicher Natur sollen durch ein solches Modell bewältigt werden. Ein Beispiel wäre der Aufbau eines Modells aus mehreren *statistischen* Methoden, um eine konkrete Umfrage sukzessiv nach verschiedenen Kriterien auszuwerten. Aus der Sicht der Bausteinebank könnte man von Bausteinmodell sprechen und darunter die Möglichkeit den unter Angabe weniger *Steuerbefehle* initiiierbaren *Aufruf* einer komplexen Folge von Prozeduren verstehen (Zusammenstellung und Aufruf eines Jobstreams).

Wird das Baukastenprinzip nicht erst für realisierte Softwarebausteine (Software-Bank), sondern schon für Problembeschreibungen (→ *Problembeschreibungstechnik*) (z.B. zur Anforderungsanalyse, Projektierung und Gewinnung von Funktionen) genutzt, erfährt die Bausteinebank eine erhebliche anwendungsorientierte Ausweitung, die über die Funktionsbeschreibung von Softwarebausteinen hinausgeht. Damit sind letztlich auch Probleme der Software-Faktoren-Bibliothek, auf ein *Projekt* bezogene Probleme der Chief-Programmer-Team-Organisation und die Problematik der Wiederauffindbarkeit und der Wiederverwendung angesprochen.

Schließlich sei noch auf den Zusammenhang mit dem Begriff Generatorsoftware eingegangen. Grundlage ist die Generalisierung anwendungsspezifischer (z.B. Lagerwesen) Fragestellungen für eine typische Klasse von Unternehmungen. Durch Parametrisierung und Modulkopplungen der geschaffenen, relativ allgemeinen Software wird eine weitgehende Anpassung an die individuellen Verarbeitungserfordernisse und *Datenstrukturen* der konkreten Unternehmung versucht. Ansätze dieser Art sind für gut strukturierbare, betriebliche Probleme größerer Unternehmungen und vor allen Dingen für die Erstellung von Software für mittlere und kleinere Unternehmungen realisiert.

Es bleibt zu erwähnen, daß das Baukastenprinzip auch für die Erstellung von Prozeßsteuerungssoftware (→ *Prozeßsteuerung¹*) und *Systemsoftware* angewandt werden kann. Eine nuancierte Bedeutung kommt dem Wort Baukastenprinzip bei der Standardsoftware zu: Viele größere Standardanwendungssoftware-Pakete, aber auch Systemsoftware, sind *hierarchisch* modular aufgebaut, so daß ein *Anwender* sich in Abhängigkeit seiner wachsenden Bedürfnisse, von einer elementaren Stufe anfangend, über mehrere Ausbaustufen sukzessive das gesamte Programmsystem anschaffen kann, ohne die alten Teile völlig austauschen oder stark ändern zu müssen. Von Aufwärtskompatibilität spricht man in diesem

Zusammenhang, wenn die alte Ausbaustufe gegen eine neue, erweiterte ausgetauscht wird, die alten Anwendungen jedoch unverändert weiterlaufen können. Dies gilt besonders für *Betriebssysteme* und auch *Hardware* (Komponenten).

Bischoff

Baukastenstruktur → *Modularität*

Baum

Teilgebiete: Datenstrukturen und Datenoperationen; Graphentheorie
tree; directed tree

Synonym: *Arboreszenz*

Eine *Datenstruktur* $B=(K,R)$ heißt (gerichteter Wurzel-) Baum, wenn R aus genau einer *Relation* besteht, die die folgenden Bedingungen erfüllt:

- 1) Es gibt genau einen *Knoten* W , der keinen *Vorgänger* hat. W heißt die *Wurzel* des Baums.
- 2) Jeder *Knoten*, mit Ausnahme der *Wurzel* W , hat genau einen *Vorgänger*.
- 3) Für jeden von der *Wurzel* verschiedenen *Knoten* k gibt es eine Folge $W=k_0, k_1, \dots, k_n=k, (n \geq 1)$ von *Knoten*, bei der k_i der *Nachfolger* von k_{i-1} ist ($1 \leq i \leq n$).

Primärdaten (→ *Basisdaten*) des Benutzers können problemadäquat baumstrukturiert sein, insbesondere werden aber *Zugriffspfade* (*Sekundärdaten*) in *Datenbanksystemen* effizient nach verschiedenen Techniken als Bäume *implementiert* (B^* -Bäume, B -Bäume). Czap

Baum, ausgeglichener

Teilgebiet: Datenstrukturen und Datenoperationen
balanced tree

Ausgeglichene *Bäume*, auch AVL-Bäume genannt, sind *binäre* Bäume, bei denen sich in jedem *Knoten* die Höhen der Teilbäume um höchstens Eins unterscheiden. Ausgeglichene Bäume mit minimaler *Knotenzahl* bei gegebener Höhe heißen *Fibonacci-Bäume*. Vollständig ausgeglichen heißt ein *binärer* Baum, wenn sich für jeden *Knoten* die Anzahl der *Knoten* seines linken und rechten Teilbaums um höchstens Eins unterscheidet. Czap

Baum, binärer

Teilgebiet: Datenstrukturen und Datenoperationen
binary tree

Ein *geordneter* Baum einer *Ordnung* ≤ 2 heißt *binärer* Baum. Entsprechend der *graphischen* Veranschaulichung unterteilt man die *Nachfolger* (soweit vorhanden) eines *Knotens* in einen linken

und rechten und nennt sie auch linken und rechten Sohn.

Binäre Bäume sind wichtig, da sich Bäume beliebiger *Ordnung* auf Bäume der *Ordnung* 2 zurückführen lassen, diese aber einfache *Speicherungs-* und *Verarbeitungsmöglichkeiten* bieten. Czap

Baum, blattorientierter

Teilgebiet: Datenstrukturen und Datenoperationen

Blattorientierte Bäume speichern *Nutzinformation* im Gegensatz zu *knotenorientierten* Bäumen nur in den *Blättern*. Die *Information* der übrigen *Knoten* dient lediglich zum Auffinden der in den *Blättern* abgelegten *Nutzinformation*. Czap

Baum, freier

Teilgebiete: Datenstrukturen und Datenoperationen; Programmierung
free tree

Ein *freier* Baum ist ein zusammenhängender *Graph* ohne *Zyklen*. Eine *Anwendung* erfolgt zur *Erfassung* (→ *Datenerfassung*) der *dynamischen* Struktur von Programmen bei einer *Darstellung* als *Flußdiagramm* (→ *Programmablaufplan*). Czap

Baum, geordneter

Teilgebiet: Datenstrukturen und Datenoperationen
ordered tree

Ein Baum $B=(K,R)$ der *Ordnung* n heißt *geordnet*, wenn die *Nachfolger* von jedem *Knoten* $k \in K$ in eine feste *Reihenfolge* gebracht (durchnumeriert) sind. Czap

Baumalgorithmus → *Ford-Algorithmus; Dijkstra-Algorithmus*

Baumautomat

Teilgebiet: Automatentheorie und formale Sprachen
tree automaton

Baumautomaten sind vergleichbar mit *endlichen* Automaten. Der wesentliche Unterschied besteht darin, daß *Baumautomaten* *Bäume* statt *Zeichenreihen* (→ *Zeichenkette*) lesen bzw. verarbeiten. Dabei ist die *Durchmusterungsstrategie* zum Besuch der einzelnen *Knoten* des Baums i.a. nicht festgelegt.

Formal ist ein *Baumautomat* ein *Quadrupel* (Q, T, d, F) , wobei

- Q eine *endliche* Menge von *Zuständen*,
- T die *Vereinigung* aller (*disjunkten*) *Eingabealphabete* T_j von *Operatoren* mit *Stelligkeit* j ,

Baumgrammatik, reguläre

- F eine Teilmenge von Q („Endzustände“) und
- d, die Menge der „Übergänge“, eine Teilmenge der Vereinigung aller Mengen $Q \times T_j \times Q^j$ (für alle Stelligkeiten j)

sind.

Ein Baumautomat „bearbeitet“ einen Eingabebaum durch Annotation der Knoten mit Zuständen gemäß d. Eine Berechnung des Baumautomaten für einen Teilbaum heißt dabei akzeptierend, wenn alle Knoten des Teilbaums markiert sind und die Wurzelmarkierung ein Endzustand ist. Partsch

Baumgrammatik, reguläre

Teilgebiet: Automatentheorie und formale Sprachen

regular tree grammar

Eine reguläre Baumgrammatik ist vergleichbar mit einer regulären (\rightarrow Menge, reguläre) Grammatik, bei der man statt terminaler Symbole *terminale Operatoren* (mit fester Stelligkeit) hat. Formal besteht eine reguläre Baumgrammatik (N, T, R, S) aus

- einer Menge N von nichtterminalen (\rightarrow nichtterminales Zeichen) Symbolen,
- einem endlichen Alphabet T von terminalen Operatoren $\sigma(i)$ (mit fester Stelligkeit i),
- einer endlichen Menge R von Regeln der Form $X \rightarrow s$ mit $X \in N$ und $s \in B(N \cup T)$ (wobei $B(N \cup T) =$ Menge der wohlgeformten Bäume über $N \cup T$),
- dem Startsymbol $S \in N$.

Ein einfaches Beispiel einer regulären Baumgrammatik ist

$(\{E, L\}, \{a(0), \text{cons}(2), \text{nil}(0)\}, \{L \rightarrow \text{nil}, L \rightarrow \text{cons}(E, L), E \rightarrow a\}, L)$,

die als Sprache die Menge der aus cons und nil gebildeten linearen Listen hat. Partsch

Baumschema

Teilgebiet: Theorie der Semantik

Programmschema, dessen unterliegende Kontrollstruktur ein Baum ist. Die Wurzel bildet die erste auszuführende Anweisung, und jede Anweisung kann von dieser auf genau einem Wege erreicht werden. Ein Baumschema ist ein Sonderfall eines Flußdiagrammschemas. Glatthaar

Bausteinebank \rightarrow Baukastenprinzip

Bausteinebank-Management-System \rightarrow Baukastenprinzip

Bausteinebibliothek \rightarrow Baukastenprinzip

Bausteineprinzip \rightarrow Baukastenprinzip

Bauteil \rightarrow Elektronik

BAZ \rightarrow Bearbeitungszentrum

B-Baum

Teilgebiet: Grundlagen von Informationssystemen

B-tree

Siehe auch: B*-Baum

Ein B-Baum ist eine baumartige Datenstruktur mit Eigenschaften, die sie speziell für die Sekundärspeicherorganisation (\rightarrow Speicherorganisation²) geeignet machen. Jeder Knoten eines B-Baums enthält im allgemeinen mehrere Einträge. Für die Struktur des B-Baums gelten die folgenden Regeln:

1. Jeder Knoten enthält höchstens $2n$ Einträge.
2. Jeder Knoten, außer der Wurzel, enthält mindestens n Einträge.
3. Jeder Knoten ist entweder ein Blattknoten (\rightarrow Blatt eines Baumes), d.h. hat keine Nachfolger, oder er hat $m+1$ Nachfolger, wobei m die Zahl seiner Einträge ist.
4. Alle Blattknoten liegen auf der gleichen Stufe. n heißt die Ordnung des B-Baums.

Munz

BCD-Code

Teilgebiet: Grundlegende Definitionen

binary coded decimal

Synonyme: Binär-Dezimal-Code; Binärcode für Dezimalziffern

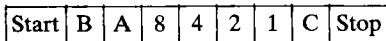
Vor ihrer Verarbeitung in Datenverarbeitungslagen müssen Zahlen in eine binäre Darstellung gebracht werden. BCD-Code wird diejenige Codierung von Dezimalzahlen (\rightarrow Dezimalsystem) genannt, bei der man nicht die Dezimalzahl insgesamt in eine binäre Darstellung umwandelt, sondern jede einzelne Dezimalziffer binär codiert. Für jede Dezimalstelle müssen mindestens vier Binärziffern (\rightarrow Binärzeichen) (Bits) bereitgestellt werden. Dies ergibt sich aus der Tatsache, daß man zur binären Darstellung der Dezimalziffern 8 und 9 vier binäre Stellen benötigt (für die Dezimalziffern 0 und 1 benötigt man eine, für 2 und 3 zwei und für 4 bis 7 drei binäre Stellen). Die bei der Verschlüsselung mit 4 Bit auftretenden Vier-Bit-Anordnungen werden Tetraden genannt; nicht benötigte Vier-Bit-Anordnungen heißen Pseudotetraden (\rightarrow Pseudodezimal).

Die binäre Codierung von Dezimalziffern mit Tetraden kann mit verschiedenen Codes erfolgen. Die folgende Tabelle enthält einige Codes (Pseudotetraden sind mit * gekennzeichnet):

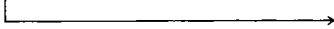
BCD-Code							Card Code	Kleinschreibung LC Char. HEX		Großschreibung UC Char. HEX	
B	A	C	8	4	2	1					
B	A					1	12-1	a	62	A	E2
B	A				2		12-2	b	64	B	E4
B	A	C			2	1	12-3	c	67	C	E7
B	A		4				12-4	d	68	D	E8
B	A	C	4		1		12-5	e	6B	E	EB
B	A	C	4	2			12-6	f	6D	F	ED
B	A		4	2	1		12-7	g	6E	G	EE
B	A		8				12-8	h	70	H	F0
B	A	C	8			1	12-9	i	73	I	F3
B		C				1	11-1	j	43	J	C3
B		C			2		11-2	k	45	K	C5
B					2	1	11-3	l	46	L	C6
B		C	4				11-4	m	49	M	C9
B			4		1		11-5	n	4A	N	CA
B			4	2			11-6	o	4C	O	CC
B		C	4	2	1		11-7	p	4F	P	CF
B		C	8				11-8	q	51	Q	D1
B			8			1	11-9	r	52	R	D2
	A	C			2		0-2	s	25	S	A5
	A				2	1	0-3	t	26	T	A6
	A	C	4				0-4	u	29	U	A9
	A		4		1		0-5	v	2A	V	AA
	A		4	2			0-6	w	2C	W	AC
	A	C	4	2	1		0-7	x	2F	X	AF
	A	C	8				0-8	y	31	Y	B1
	A		8			1	0-9	z	32	Z	B2
		C	8		2		0	0	15)	95
						1	1	1	02	=	82
					2		2	2	04	□	84
		C			2	1	3	3	07	;	87
			4				4	4	08	:	88
		C	4		1		5	5	0B	%	8B
		C	4	2			6	6	0D	'	8D
			4	2	1		7	7	0E	"	8E
			8				8	8	10	.	90
		C	8			1	9	9	13	(93
B	A	C					12	&	61	+	E1
B	A		8		2	1	12-3-8	⊙	76	.⊙	F6

BCD-Code

BCD-Code							Card Code	Kleinschreibung LC		Großschreibung UC	
B	A	C	8	4	2	1		Char.	HEX	Char.	HEX
B							11	-Ⓝ	40	-Ⓝ	C0
B		V	8		2	1	11-8-3	\$	57	!	D7
	A						8-4	@	20	¢	A0
	A	C	8		2	1	0-8-3	,	37	,	B7
	A	C					0-1	/	23	?	A3
			8		2	1	8-3	#ⓉEOA	16	±ⓉEOA	96
		C	8	4	2	1	9-7	ⓄEOT	1F	ⓄEOT	9F
	A	C	8	4	2		0-9-6	ⓅEOB	3D	ⓅEOB	BD
		C					Leerspalte	Space	01	Space	81
	B		C	8	4	2	11-9-6	Backspace	5D	Backspace	DD
			8	4	2		9-6	Upshift	1C	Upshift	9C
B	A		8	4	2		12-9-6	Downshift	7C	Downshift	FC
B		C	8	4		1	11-9-5	CR, LF	5B	CR, LF	DB
B			8	4	2	1	11-9-7	Idle	5E	Idle	DE
B	A		8	4		1	12-9-5	Tab	7A	Tab	FA
B	A	C	8	4	2	1	12-9-7	Delete	7F	Delete	FF



BCD Code auf dem Übertragungsweg



Erstes Bit eines Zeichens auf der Leitung

Der Begriff BCD-Code ist mehrdeutig. Er bezeichnet speziell den 8-4-2-1-Code und im weiteren Sinne alle Binärcodes für Dezimalziffern. Beim 8-4-2-1-Code werden die Dezimalziffern im Dualsystem, d.h. als Dualzahlen, codiert. Die Wertigkeit der Bits in den Tetraden beträgt demnach von links nach rechts 8,4,2,1. Beim Aiken-Code liegt eine andere Wertigkeit der vier Bit-Positionen vor. Wie auch der Exzeß-3-Code vereinfacht er die beim Rechnen in dieser Darstellung notwendigen Prüfvorgänge und erleichtert die Komplementbildung.

Beim BCD-Code im weiteren Sinne werden zur Darstellung von Dezimalzahlen mehr Bit-Stellen benötigt als bei rein dualer Codierung. Bei Datenverarbeitungsanlagen mit 8-Bit-Speicherplätzen (→ Speicherzelle) (Bytes) können mit gewissen Einschränkungen zwei Tetraden pro Speicherplatz untergebracht werden (gepackte Zifferndarstellung). Für diese Anlagen wurde der BCD-Code zum EBCDI-Code (extended binary coded decimal) erweitert. Der erweiterte Code

Tetraden	BCD-Code (8-4-2-1 Code)	Aiken-Code (2-4-2-1-Code)	Excess-1-Code
0000	0	0	*
0001	1	1	*
0010	2	2	*
0011	3	3	0
0100	4	4	1
0101	5	*	2
0110	6	*	3
0111	7	*	4

1000	8	*	5
1001	9	*	6
1010	*	*	7
1011	*	5	8
1100	*	6	9
1101	*	7	*
1110	*	8	*
1111	*	9	*

----- = Symmetrieachse

Tabelle: Vergleich verschiedener Codes für Ziffern.