

Algorithmen und algorithmische Sprachkonzepte

Kapitel 3 in Küchlin/Weber: Einführung in die Informatik

Was ist ein Programm?

Programm [gr.], Reihenfolge, Arbeitsplan; Zielsetzung

Programmierung, die Aufstellung eines *Programms*, d.h. einer (meist) festgelegten **Aufeinanderfolge von Befehlen** (Anweisungen, oft in einer Programmiersprache), die zur automatischen Behandlung u. Lösung des so programmierten Problems mit Hilfe eines Automaten führt

Herders Neues Volkslexikon, 1983

- **Programm** im **allgemeinen**
 - Fernsehprogramm, Konferenzprogramm, Parteiprogramm
- Programm im **speziellen**
 - Programm eines Computers (auch: Waschmaschine, Videorecorder)

Was ist ein Programm?

Programm [griech. >schriftliche Bekanntmachung<, >Tagesordnung<]
Ankündigung, Plan, Ziel; festgelegte Folge (z.B. bei einer Veranstaltung oder in Fernsehen und Rundfunk); Ankündigungszettel; Warenangebot . . .

. . . 1) vollständige Anweisung für automatisch gesteuerte Maschinen wie Werkzeug- oder Textilmaschinen, Rechenanlagen u.a., nach der die zur Bearbeitung einer Aufgabe erforderlichen Arbeitsgänge ablaufen sollen

Programmierung, Datenverarbeitung: das Erstellen eines Programms.

Programmiersprache, Datenverarbeitung: zum Abfassen von Programmen für digitale Rechenanlagengeschaffene Sprache.

dtv-Lexikon in 20 Bänden, Bd. 14, 1999

Was ist ein Programm?

"Definitionen" des Begriffs 'Programm' in informatiknahen Quellen:

Programm heißt ein mit Hilfe eindeutiger Regeln formulierter **Algorithmus**, der in dieser Form von einem bestimmten Rechner übernommen und ausgeführt werden kann.

dtv-Atlas zur Informatik, 1995

Um einen **Algorithmus** in einer von einer Maschine ausführbaren Form beschreiben zu können, verwendet man eine formale Sprache zur Repräsentation von Algorithmen. Eine Beschreibung eines **Algorithmus** heißt ein **Programm**, die formale Sprache eine **Programmiersprache**.

Manfred Broy, Informatik, Teil I, Springer, 1992

Nach R. Manthey, Vorlesung Informatik I, Universität Bonn, WS 2001/2002

- **Begriffsdefinition**

- Ein **Algorithmus** (algorithm) ist die Beschreibung eines Verfahrens, um aus gewissen Eingabegrößen bestimmte Ausgabegrößen zu berechnen.

Dabei müssen folgende Bedingungen erfüllt sein

- **Spezifikation**
- **Durchführbarkeit**
- **Korrektheit**

- **Spezifikation** (**präzise** Beschreibung)
 - **Eingabespezifikation**
 - Welche Eingabegrößen sind erforderlich?
 - Welchen Anforderungen müssen diese Größen genügen?
 - **Ausgabespezifikation**
 - Welche Ausgabegrößen (Resultate)?
 - Welche Eigenschaften haben diese?

- **Durchführbarkeit**

- **Endliche Beschreibung**

Das Verfahren muss in einem endlichen Text vollständig beschrieben sein

- **Effektivität**

Jeder Schritt des Verfahrens muss effektiv (d.h. tatsächlich) „mechanisch“ ausführbar sein (NB: nicht gleichbedeutend mit „Effizienz“ Wirtschaftlichkeit)

- **Determiniertheit**

Der Ablauf ist zu jedem Zeitpunkt fest vorgegeben

- **(Partielle) Korrektheit**

Für jede Eingabe, die der Eingabespezifikation genügt, wird eine Ausgabe berechnet, die der Ausgabespezifikation genügt

Sichtweise:

- **Kontrakt**
- Wenn-dann-Beziehung

- **Totale Korrektheit**
 - Partielle Korrektheit
und zusätzlich
 - **Terminierung**
Für jede Eingabe, die der Eingabespezifikation genügt, hält der Algorithmus nach endlich vielen Schritten an

- **Bemerkung**

Nach dieser Begriffsbestimmung gäbe es also keine

- **nicht-deterministische**
- **nicht-terminierende**
- ...

Algorithmen

- Diese Begriffe werden durchaus verwendet!
Aber hier (zunächst) nicht.

Abstrakte Maschinenmodelle

- Entwicklung von **Algorithmen** kann weitgehend **ohne** ein **konkretes Maschinenmodell** erfolgen
- Auch (moderne, höhere) **Programmiersprachen** sind so entworfen, dass Programme auf **verschiedenen Computern** ablaufen können
- Trotzdem fließen Eigenschaften realer Computer auch in das Design von Programmiersprachen ein
 - Programme sollen nicht nur **effektiv** durchführbar sein, sondern auch **effizient**!

Ein "Algorithmus" aus dem täglichen Leben

The screenshot shows the Netscape browser window displaying the General-Anzeiger Online website. The address bar shows the URL <http://www.general-anzeiger-bonn.de/index4.html>. The website features a navigation menu on the left with categories like NEWS, SPORT, KULTUR, WIRTSCHAFT, WETTER, FREIZEIT, ANNONCEN, RAT+HILFE, DIE REGION, ABO-SERVICE, GA-PLUS, and DIALOG. The main content area is titled 'General-Anzeiger online' and 'Lauter Lieblingsgerichte'. A red box highlights the recipe 'Lachs im Spinatmantel' (Zugrid Geisler), which includes the ingredients and preparation steps. A blue box on the right states 'Rezepte sind "Kochalgorithmen" !'. A red arrow points from the 'Eingabespezifikation' label to the ingredients list, and a blue arrow points from the 'Rezepte sind "Kochalgorithmen" !' box to the preparation steps.

Eingabespezifikation

Rezepte sind "Kochalgorithmen" !

Lachs im Spinatmantel
(Zugrid Geisler)

Zutaten für 4 Personen: (Zubereitungszeit: ca. 25 Minuten) 300 g Lachsfilet, etwas Zitronensaft, 8 große Spinatblätter, Salzwasser, 1 kleine Möhre, 1/2 Zucchini, Salz, weißer Pfeffer, 1 Esslöffel geschmolzene Butter, 1 Teelöffel gehacktes Estragon, 1 Esslöffel gehackte Petersilie

Zubereitung: Lachsfilet in acht Stücke schneiden und mit Zitronensaft beträufeln. Spinatblätter 30 Minuten in Salzwasser blanchieren. Die Lachsstücke auf die Blätter verteilen. Möhre und Zucchini in sehr feine Streifen schneiden. Über den Lachs geben. Mit Salz und Pfeffer würzen. Geschmolzene Butter darauf träufeln, Kräuter darüber streuen. Lachsstücke in die Blätter einwickeln. Auf einem Sieb über heißem Dampf etwa zehn Minuten garen.

[zum Inhalt](#)

Ein "Algorithmus" aus dem täglichen Leben

- Lachs im Spinatmantel - für 4 Personen (**Ausgabespezifikation**)
- Zutaten (**Eingabespezifikation**):
300 g Lachsfilet, etwas Zitronensaft, 8 große Spinatblätter, Salzwasser, 1 kleine Möhre, 1/2 Zucchini, Salz, weißer Pfeffer, 1 Esslöffel geschmolzene Butter, 1 Teelöffel gehacktes Estragon, 1 Esslöffel gehackte Petersilie
- Zubereitung (**Beschreibung des Algorithmus**):
Lachsfilet in acht Stücke schneiden und mit Zitronensaft beträufeln. Spinatblätter 30 Minuten in Salzwasser blanchieren. Die Lachsstücke auf die Blätter verteilen. Möhre und Zucchini in sehr feine Streifen schneiden. Über den Lachs geben. Mit Salz und Pfeffer würzen. Geschmolzene Butter darauf träufeln, Kräuter darüber streuen. Lachsstücke in die Blätter einwickeln. Auf einem Sieb über heißem Dampf etwa zehn Minuten garen.

Eine sehr alte Beschreibung eines Algorithmus

Verdoppeln einer ganzen Zahl nach **Adam Riese** (16. Jahrhundert)

"Duplieren

Lehret wie du ein zahl zweyfaltigen solt. Thu ihm also:
Schreib die zahl vor dich / mach ein Linien darunter /
heb an zu forderst / Duplir die erste Figur. Kompt ein zahl
die du mit einer Figur schreiben magst / so setz die unden.
Wo mit zweyen / schreib die erste / Die ander behalt im sinn.
Darnach duplir die ander / und gib darzu / das du behalten
hast / und schreib abermals die erste Figur / wo zwo vor-
handen / und duplir fort biß zur letzten / die schreibe gantz
auß / als folgende Exempel aufweisen.

41232
82464

98765
197530

68704
137408

"

aus: A. Risen, Rechenbuch, 1574

Herkunft des Wortes „Algorithmus“

Muhammad ibn Musa abu Djafar **al-Choresmi**

ca. 780-850 n.Chr.

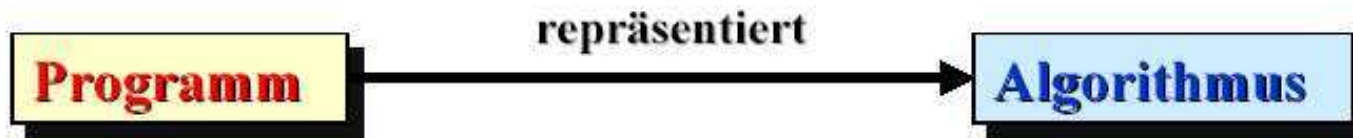
- arabischer Mathematiker, geboren in Choresmien (heute Usbekistan)
- lebte und wirkte in Bagdad im "Haus der Weisheit"
- war beteiligt an der Übersetzung der Werke griechischer Mathematiker ins Arabische
- schrieb ein "Kurzgefasstes Lehrbuch für die Berechnung durch Vergleich und Reduktion"
- lateinische Übersetzung dieses Buches ("**liber algorismi**") kam durch Kreuzfahrer nach Europa
- verfasste auch ein Buch mit dem Titel "Al-Mukhtasar fi Hisab **al-Jabr** wa l-Muqabala"

Algebra

Algorithmus

Algorithmen und Programme

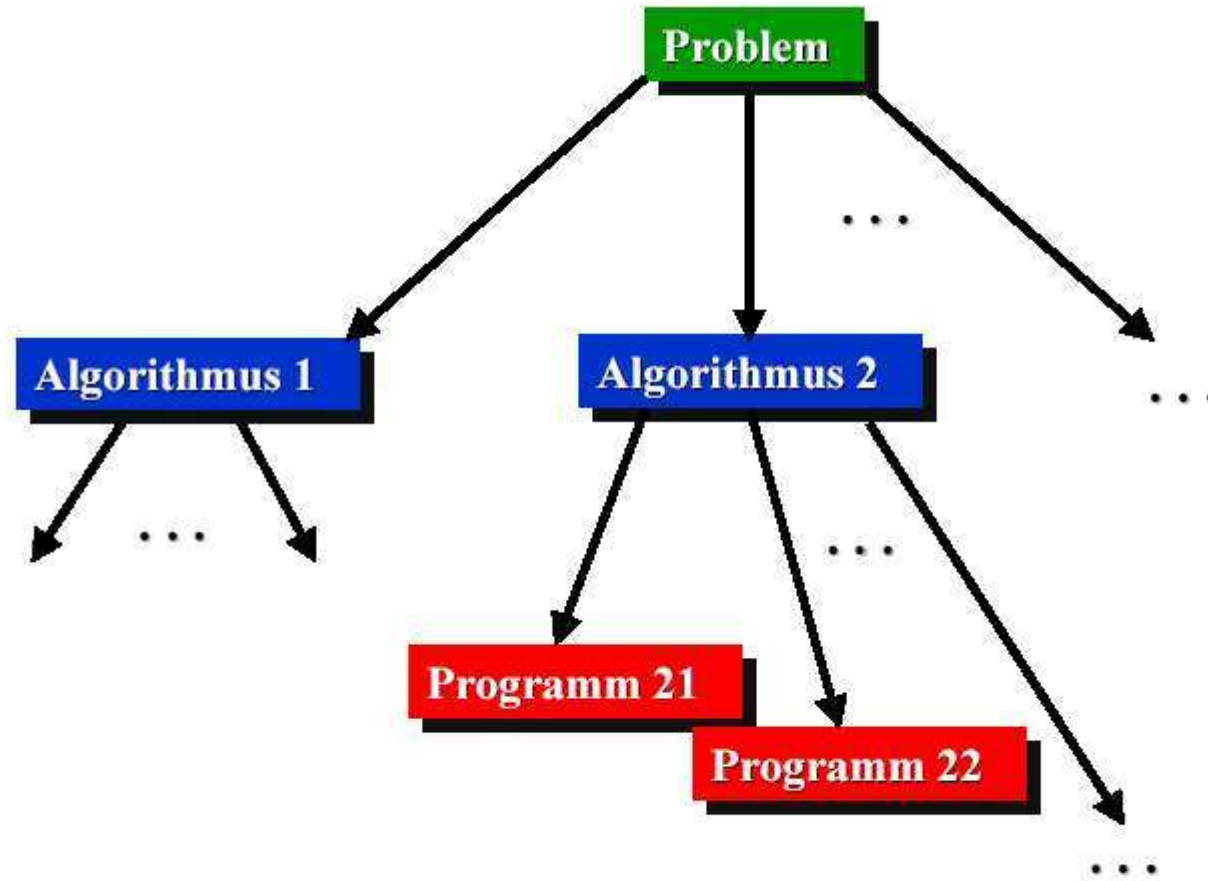
- grundlegender Zusammenhang zwischen den zentralen Begriffen:



- Jedes Programm repräsentiert einen bestimmten Algorithmus.
- Ein Algorithmus wird durch viele verschiedene Programme repräsentiert.
- Programmieren setzt Algorithmenentwicklung voraus:

Kein Programm ohne Algorithmus !

Problem – Algorithmus – Programm



Nach R. Manthey, Vorlesung Informatik I, Universität Bonn, WS 2001/2002

- **Effizienz** ist weiteres wichtiges **Prinzip**, das beim Entwurf zu beachten ist (neben den Bedingungen der Begriffsdefinition):
- Der Algorithmus soll möglichst wenig **Aufwand** verursachen
 - **Rechenschritte** (Zeit)
 - **Speicherbedarf**

- Frage der **Komplexität** (Aufwand relativ zur Größe der Eingabe) ist wichtiges Thema der Theoretischen Informatik
- Komplexität
 - von Algorithmen
 - von Problemklassen (die Komplexität, die die besten Algorithmen für die Problemklasse haben)

- **Korrektheit** von Algorithmen bzw. Programmen ist eine der **wichtigsten Problemstellungen** der Informatik
 - **Konstruktion** korrekter Algorithmen bzw. Programme
 - Nachweis der Korrektheit (**Validierung**)
 - Formaler Beweis der Korrektheit (**Verifikation**)
 - Finden von Fehlern (insbes. durch **Testen**)

Beschreibung von Algorithmen

- Viele Möglichkeiten
 - **Alltagssprache**
 - Konkrete **Programmiersprache**
 - Dazwischen: Vielzahl von Notationen, die den Übergang zwischen Problembeschreibung und Programm erleichtern
 - **Flussdiagramme**
 - **Pseudocode**

Strukturelemente von Algorithmen

- **Hintereinanderausführung** einzelner Schritte oder ganzer Teilalgorithmen
- **Verzweigung** (bedingte Ausführung von Teilen)
- **Wiederholung**
 - **Iteration** (Schleife)
Beispielsweise
 - „folgendes solange wiederholen bis ...“
 - „zurück zu Schritt (2)“
 - **Rekursion**
Algorithmus benutzt sich selbst für (einfacheres) Teilproblem

Grundschema des Algorithmenaufbaus

- Folgendes **Grundschema** wird uns bei vielen Algorithmen begegnen

Grundschema des Algorithmenaufbaus

	Name des Algorithmus und Liste der Parameter Spezifikation des Ein-/Ausgabeverhaltens
Schritt 1	Vorbereitung: Einführung von Hilfsgrößen etc.
Schritt 2	Trivialfall? Prüfe, ob ein einfacher Fall vorliegt. Falls ja, Beendigung mit Ergebnis.
Schritt 3	Arbeit (Problemreduktion, Ergebnisaufbau): Reduziere die Problemstellung X auf eine einfachere Form X' , mit $X > X'$ bezüglich einer wohlfundierten Ordnung $>$. Baue entsprechend der Reduktion einen Teil des Ergebnisses auf.
Schritt 4	Rekursion bzw. Iteration: Rufe zur Weiterverarbeitung den Algorithmus mit dem reduzierten X' erneut auf (Rekursion), bzw. fahre mit X' anstelle X bei Schritt 2 fort (Iteration). \square

Grundschema des Algorithmenaufbaus: Beispiel

- Algorithmus zur Berechnung des Rests r der **Ganzzahldivision** a/b
- Eingabespezifikation:
Ganze Zahlen a und b mit $0 \leq a$ und $0 < b$
- Ausgabespezifikation:
Ganze Zahl r , so dass
 - $0 \leq r < b$
 - es gibt ganze Zahl c mit $cb + r = a$

Grundschema des Algorithmenaufbaus: Beispiel

mod(a, b)

// Anforderungen:

// $a, b \in \mathbb{Z}, a \geq 0, b > 0$.

// Zusicherung:

// Das Resultat ist der Rest der Division a/b .

1. Kopiere a nach r .
2. Prüfe, ob r größer oder gleich b ist.
3. Falls nein, gib das Resultat r aus.
4. Falls ja, ziehe von r den Wert b ab (und speichere das Resultat in r).
5. Mache weiter mit Schritt 2.