

6 Fundamentale Ideen der Informatik

Jede Wissenschaft muß sich immer wieder fragen, was ihre unveränderlichen Inhalte sind, aus denen sie ihr Selbstverständnis bezieht. In der kurzen Geschichte der Informatik hat sich bereits eine Reihe von typischen Methoden und Arbeitsweisen herausgebildet, die man als *fundamentale Ideen* bezeichnet. Diese Ideen besitzen gewisse Merkmale, die sie für den Forschungs- und vor allem für den Lehr- und Lernprozeß besonders geeignet erscheinen lassen.

Als **fundamentale Ideen** bezeichnen wir Denk-, Handlungs-, Beschreibungs- oder Erklärungsschemata, die

- (1) in verschiedenen Gebieten der Informatik vielfältig anwendbar oder erkennbar sind,
- (2) auf jedem intellektuellen Niveau aufgezeigt und vermittelt werden können,
- (3) zur Annäherung an eine gewisse idealisierte Zielvorstellung dienen, die jedoch faktisch möglicherweise unerreichbar ist,
- (4) in der historischen Entwicklung des Bereichs deutlich wahrnehmbar sind und längerfristig relevant bleiben,
- (5) einen Bezug zu Sprache und Denken des Alltags und der Lebenswelt besitzen.

Für Lernende, insbesondere Anfänger, hat das Wissen um die fundamentalen Ideen unschätzbare Vorteile:

- Man kann viele Phänomene, Methoden und Inhalte der Informatik, die in zahlreichen Veranstaltungen des Studiums weitgehend isoliert voneinander vermittelt werden, miteinander verknüpfen, indem man sie unter eine oder mehrere Ideen subsumiert und so gedanklich ein in sich geschlossenes Informatikgebäude errichtet.
- Die Ideen können dazu beitragen, zwischen aufeinander aufbauenden Veranstaltungen des Studiums den roten Faden (wieder) zu erkennen, und man kann verfolgen, wie gewisse Inhalte entlang dieses Fadens immer wieder aufgegriffen, erweitert und schließlich einer theoretischen und praktischen Analyse in voller Breite und Tiefe unterzogen werden.
- Man kann für viele Informatik-elemente, die in einem langwierigen wissenschaftlichen Prozeß ihre heutige formale Form gewonnen und damit möglicherweise zugleich ihren illustrativen Gehalt verloren haben, wieder lebensweltliche (im Alltag sichtbare, anschauliche) Vorstufen entdecken.

In dieser Vorlesung werden einige der im folgenden genannten Ideen nur rudimentär oder gar nicht erklärt. Dies bleibt weiterführenden Vorlesungen vorbehalten. Es empfiehlt sich, den Ideenkatalog während des Studiums regelmäßig zur Hand zu nehmen, zu verinnerlichen und zu beobachten, welche Ideen in der aktuellen Veran-

staltung erstmalig oder vertiefend behandelt werden oder wie sich die betrachteten Inhalte in die Ideenkollektion einordnen.

Die Ideenkollektion.

Unter den fundamentalen Ideen der Informatik stechen drei besonders hervor, weil sie, gewissermaßen als übergreifende Ideen, in allen Bereichen der Informatik eine herausragende Rolle spielen: Die Ideen der *Algorithmisierung*, der *Sprache* und der *strukturierten Zerlegung*. Diese Ideen wollen wir im folgenden genauer analysieren, da sie Anhaltspunkte für eine Reihe weiterer Ideen liefern.

Algorithmisierung.

Über die Bedeutung der Algorithmisierung als fundamentale Idee der Informatik besteht weitgehend Konsens. Mit dieser Idee verbindet sich die Zielvorstellung, alle Probleme ließen sich durch maschinell nachvollziehbare Verfahren, deren Korrektheit jederzeit gesichert ist, effizient lösen.

Eine genauere Analyse der Aktivitäten beim Algorithmisieren liefert eine Fülle weiterer Ideen. Wir unterscheiden vier große Bereiche: Beim Entwurf eines Algorithmus bedient man sich häufig bestimmter Grundmuster, sog. *Entwurfsparadigmen*, wie *Divide-and-Conquer*, *Backtracking*. Diesen Entwurf setzt man anschließend in ein Programm um. Hierfür stehen gewisse universelle Konzepte zur Verfügung, um den Daten- und Kontrollbereich zu beschreiben, wie z.B. *Konkatenation*, *Alternative*, *Rekursion* usw. Das fertige Programm wird auf einem (oder mehreren) Prozessoren ausgeführt. Damit verbunden ist die Idee des *Prozesses*. Die letzte Ideengruppe im Rahmen der Algorithmisierung befaßt sich mit der Bewertung von Algorithmen unter Qualitätsgesichtspunkten. Die beiden zentralen Bewertungskriterien sind *Korrektheit* und *Komplexität*, jeweils mit einer Reihe weiterer zugehöriger Ideen.

Sprache.

Nicht nur bei der Programmierung (Programmiersprachen), bei der Spezifikation (Spezifikationssprachen), bei der Korrektheitsbeweisen (Logikkalküle), in Datenbanken (Anfragesprachen), bei Betriebssystemen (Kommandosprachen) spielt die Idee der Sprache eine herausragende Rolle, vielmehr besteht in der Informatik eine generelle Tendenz zur Versprachlichung von Sachverhalten. Dies gilt auch für Bereiche, bei denen zunächst kein unmittelbarer Bezug zu einer sprachlichen Darstellung erkennbar ist, z.B. beim VLSI-Entwurf oder bei der Bildverarbeitung. Dieses Vorgehen bietet u.a. folgenden Vorteil: Es vereinheitlicht die Sichtweise, denn jedes Problem reduziert sich dann auf ein

Problem über Wörtern; die algorithmische Manipulation von Sprachen und Wörtern ist andererseits gut erforscht.

Mit der Idee der Sprache verbindet sich die Zielvorstellung, daß sich alle informatik-relevanten Sachverhalte durch gewisse Zeichenfolgen beschreiben lassen, die nach einfachen Bildungsgesetzen aufgebaut sind und die untereinander auf effiziente algorithmische Weise semantikerhaltend transformierbar sind.

Beispiel: Veranschaulichung einer Rechnerarchitektur durch ein Ebenenmodell (s. Abschnitt 5).

In engem Zusammenhang mit Sprachen stehen die Ideen von *Syntax* und *Semantik*, ferner die verschiedenen Ansätze zur semantikerhaltenden Transformation von Wörtern einer Sprache in eine andere, z.B. die Ideen der *Übersetzung*, *Interpretation* oder *operationalen Erweiterung*.

Strukturierte Zerlegung.

Bei der strukturierten Zerlegung unterscheiden wir zwei zueinander orthogonale Aspekte:

- Der vertikale Aspekt, die *Hierarchisierung* (Abb. 1), beschreibt die Zerlegung eines Gegenstands in aufeinander aufbauende Ebenen unterschiedlichen Abstraktionsniveaus. Geleitet wird diese Vorgehensweise von der Zielvorstellung einer schrittweisen totalen Zerlegbarkeit jedes Systems in eine endliche Folge von Hierarchieebenen, die einen unterschiedlichen Abstraktionsgrad besitzen, aber semantisch äquivalent sind und algorithmisch ineinander überführt werden können.

Die Idee der Hierarchisierung finden wir u.a. in folgenden Zusammenhängen: Ebenenmodelle der Rechnerarchitektur (s.o.), Sprachhierarchien (z.B. die Chomsky-Hierarchie), Maschinenmodelle, Komplexitäts- und Berechenbarkeitsklassen, virtuelle Maschinen, ISO-OSI-Ebenenmodell.

- Der horizontale Aspekt, die *Modularisierung* (Abb. 2), beschreibt die Zerlegung eines Gegenstands in einzelne Teile gleichen Abstraktionsniveaus. Hinter der Modularisierung verbirgt sich die Zielvorstellung, jedes System ließe sich durch die Eigenschaften seiner Teile vollständig erklären und als Summe total voneinander unabhängiger Teile (bottom-up) auffassen bzw. in total voneinander unabhängige Teile zerlegen (top-down). Diese Zielvorstellung ist zwar faktisch unerreichbar, denn die Eigenschaften eines Systems lassen sich nicht nur aus den Eigenschaften der Subsysteme allein ableiten, sondern sie werden auch durch ihre Beziehungen untereinander beeinflusst; die normative Funktion dieser Idee äußert sich jedoch in dem Bestreben, die Abhängigkeiten zwischen den Teilsystemen (Schnittstellen) dann wenigstens so gering wie möglich zu halten.

Die hierarchische Modularisierung entsteht als Mischform aus diesen beiden Grundformen (Abb. 3).

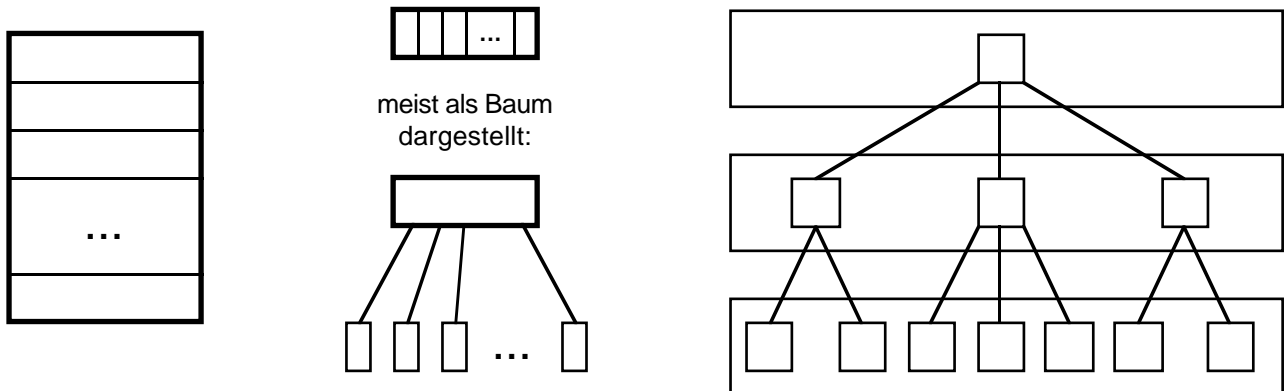


Abb. 1: Hierarchisierung

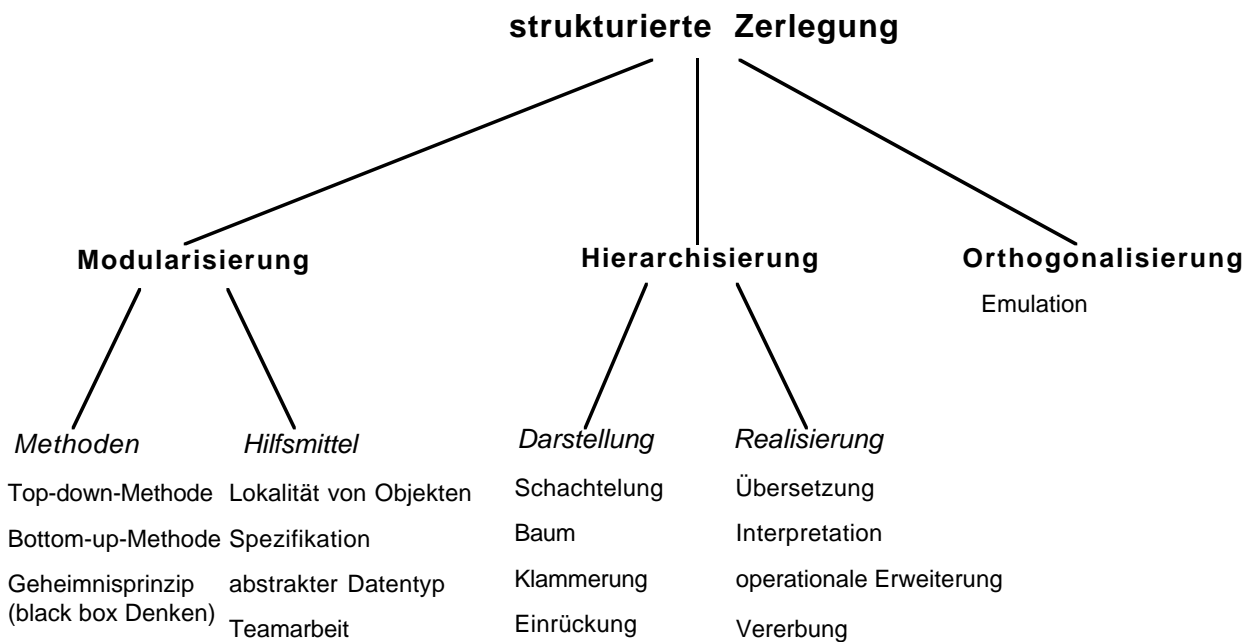
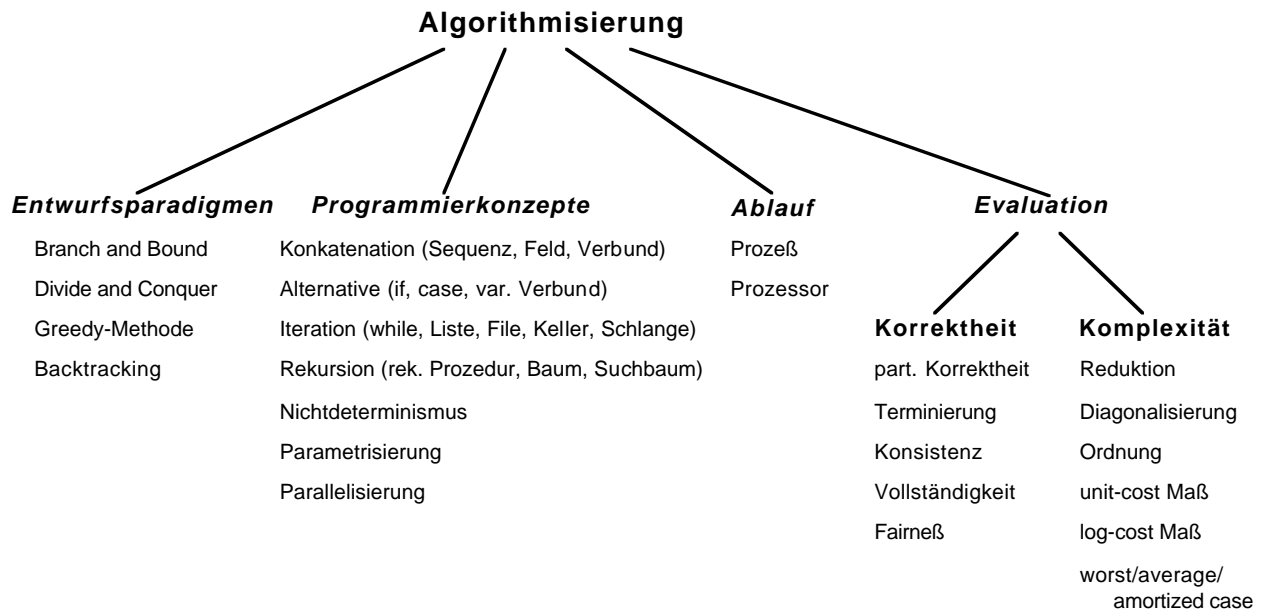
Abb. 2: Modularisierung

Abb. 3: Hierarchische Modularisierung

Offenbar kommt jeder Zerlegungsprozeß irgendwann zum Ende, spätestens dann, wenn ein atomares Niveau erreicht wird. Diese Beobachtung führt auf die Idee der Erzeugendensysteme, wir wollen in Anlehnung an eine ähnliche Operation in der linearen Algebra von der Idee der *Orthogonalisierung* sprechen. Unter Orthogonalisierung eines Objektbereiches verstehen wir die Angabe einer möglichst kleinen Zahl von Basiselementen des Bereiches sowie von Operationen auf dieser Basis, so daß jedes beliebige andere Objekt des Bereiches durch endliche Anwendung der Operationen aus den Basiselementen erzeugt werden kann. Hinter dieser Idee verbirgt sich der Wunsch nach "Einfachheit" und die reduktionistische Zielvorstellung, jeder Objektbereich ließe sich in der beschriebenen Weise orthogonalisieren, was für viele in der Informatik vorkommenden Bereiche auch tatsächlich sehr häufig zutrifft.

Zum Nachweis der Nicht-Orthogonalität verwendet man üblicherweise die Idee der *Emulation*: Gegeben sei ein Erzeugendensystem. Kann man eines der Elemente des Systems durch die übrigen darstellen, so ist das System nicht orthogonal.

Abb. 4 zeigt den vollständigen Katalog fundamentaler Ideen der Informatik. Er enthält alle bisher genannten Ideen, thematisch gruppiert und hierarchisch (da ist die Idee wieder!) strukturiert. Einige noch nicht genannte Ideen runden die einzelnen Gruppen ab. Übergeordnete Ideen sind wie erwähnt Algorithmisierung, strukturierte Zerlegung und Sprache. Man beachte: Bei den kursiv dargestellten Bezeichnungen handelt es sich um Oberbegriffe für Ideengruppen, die zur Systematisierung hinzugenommen wurden, und nicht um Ideen.



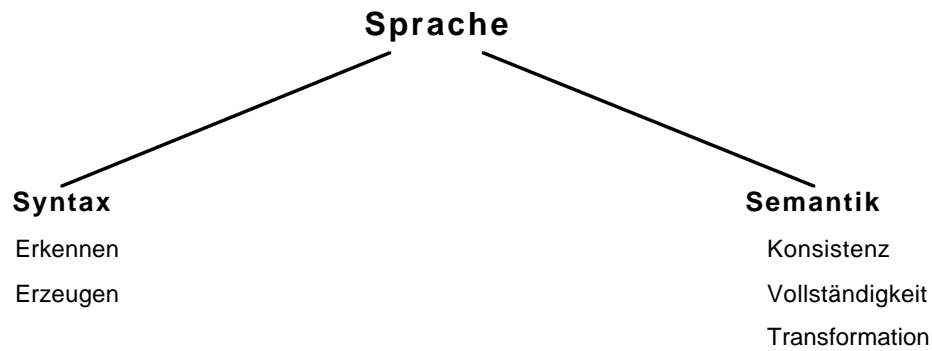


Abb. 4: Fundamentale Ideen der Informatik

Man kann sich nun fragen, warum gerade die drei Ideen Algorithmisierung, Sprache und strukturierte Zerlegung das Fundament der Informatik bilden.

Eine plausible Erklärung stützt sich auf die Modellbildungsaktivitäten in der Informatik: Bei der Suche nach Antworten auf die Frage "Was ist Informatik?" werden häufig Methoden der informatischen Modellbildung betont. Gelegentlich wird die Informatik auch als *die Wissenschaft von der Modellbildung* bezeichnet. Akzeptiert man diese Charakterisierung, so lassen sich die drei Masterideen als tragende Säulen der Modellbildung auffassen (Abb. 5):

- Mit der *strukturierten Zerlegung* sind die Ideen verbunden, mit deren Hilfe man die reale Welt analysiert und die modellrelevanten Eigenschaften ableitet.
- Das Modell wird anschließend auf der Basis einer Beschreibungssprache präzisiert und öffnet sich so weiteren syntaktischen und vor allem semantischen Analysen und Transformationen.
- Der dynamische Aspekt von Modellen, die Möglichkeit, sie zu simulieren, wird durch die *Algorithmisierung* erfaßt. Die zugehörigen Ideen dienen dem Entwurf und dem Ablauf von Simulationsprogrammen, wobei die Simulation im weitesten Sinne zu verstehen ist.

Informatik präsentiert sich aus dieser Sicht also als

Wissenschaft von der Modellierung künstlicher Welten.

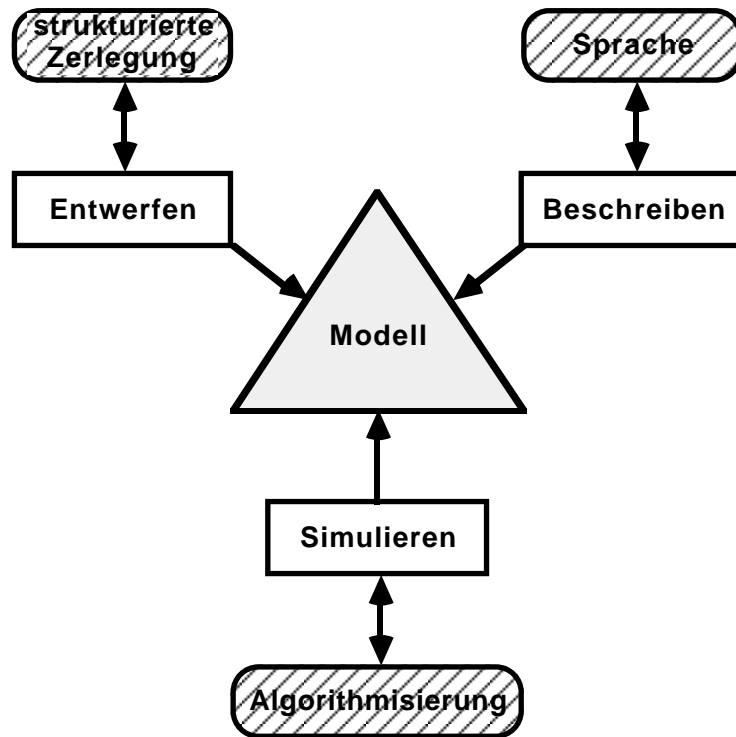


Abb. 5: Phasen der Modellbildung