

Systemtheorie 1

Formale Systeme 1

#342234

<http://fmv.jku.at/fs1>

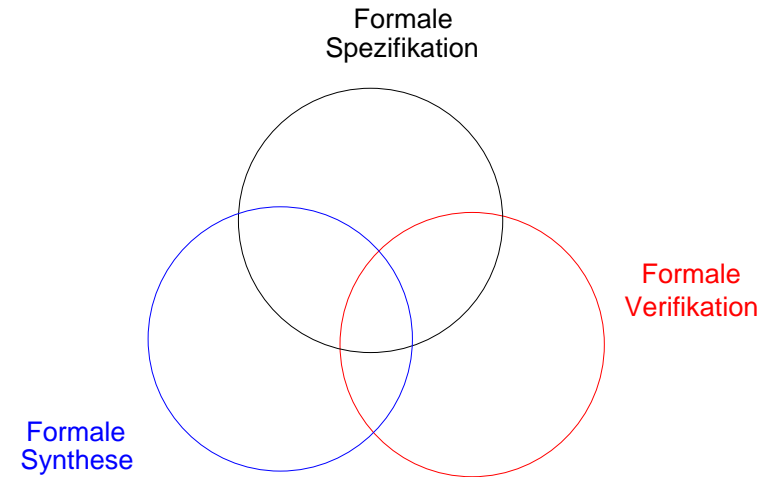
WS 2006/2007

Johannes Kepler Universität

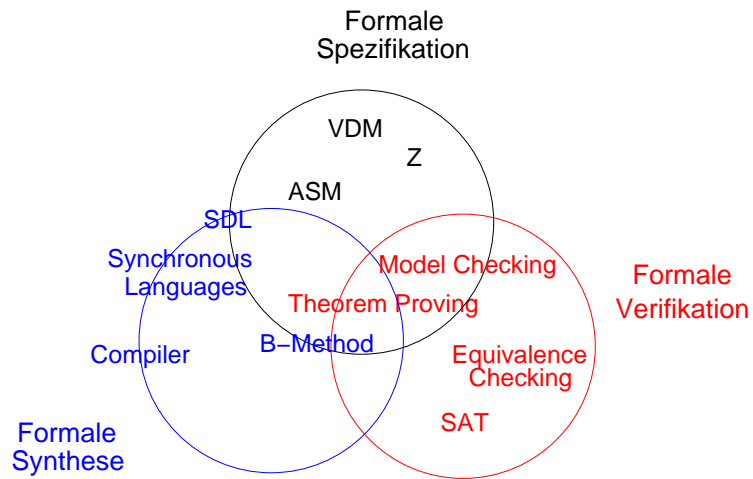
Linz, Österreich

Univ. Prof. Dr. Armin Biere

Institut für Formale Modelle und Verifikation

<http://fmv.jku.at>

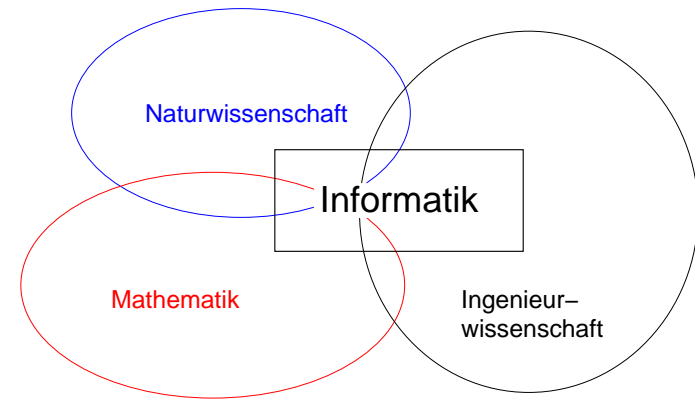
Spezifische Formale Methoden



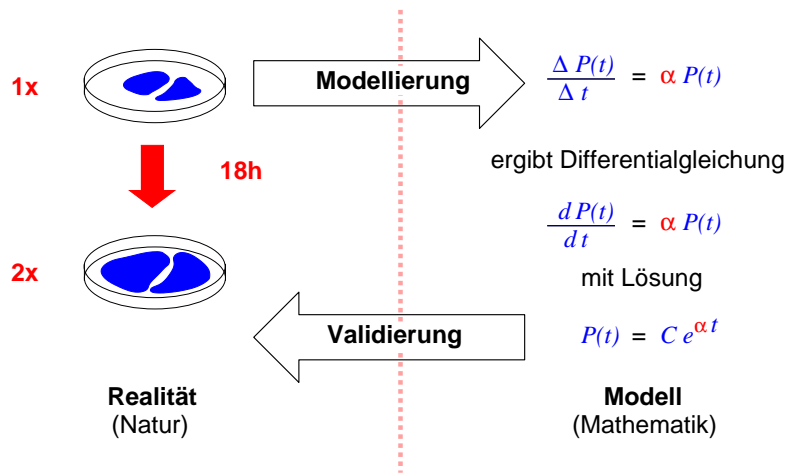
Einordnung der Vorlesung

- Formale Grundlagen III (4. Semester)
 - Formale Modellierung von Informatik-Systemen (meist diskret)
- Systemtheorie I (Formale Systeme I, 5. Semester)
 - Algorithmische Aspekte der Verifikation inkl. Simulation im Allgemeinen
- Systemtheorie II (Formale Systeme II, Model Checking, > 5. Semester)
 - Spezifische Algorithmische Aspekte des Model Checking mit SAT

- Formale Methoden als Rückrat der Informatik
 - Abstraktion ist das Kern-Geschäft eines Informatikers
 - Informatik-Systeme **sind** mathematische Gebilde
 - Abstraktionen stellen sich als formale Modelle dar
- Formale Technologien breiten sich in der Praxis aus
 - Simulation abstrakter Modelle zur Validierung oder Optimierung
 - Beispiel: tagtäglicher Gebrauch von Equivalence Checking beim Chip-Design



Größenwachstum einer Bakterienpopulation



- Modellierung und Auswertung von Computer-Modellen
 - von Phänomenen der Natur- aber auch Ingenieurwissenschaften
 - erlaubt Projektion in die Zukunft ...
z.B. [Wettervorhersage](#)
 - ... und Optimierung, wenn der Mensch eingreifen kann
z.B. [Umkehr der Klima-Erwärmung durch weniger Kohlendioxid-Ausstoß](#)
- Beschreibung der Zusammenhänge wiederum durch Gleichungen
 - Gleichungen können i.Allg. **nicht geschlossen** gelöst werden
 - Validierung durch Simulation mit numerischen Methoden
- **Computational Science als Teil der Informatik (!?)**

am Beispiel Cocomo nach [Boehm81]

Wieviel kostet die Herstellung eines Programmes einer bestimmter Größe?

$$\text{Applikations-Programme: PM} = 2.4 \cdot (\text{KDSI})^{1.05}$$

$$\text{Hilfs-Programme: PM} = 3.0 \cdot (\text{KDSI})^{1.12}$$

$$\text{System-Programme: PM} = 3.6 \cdot (\text{KDSI})^{1.20}$$

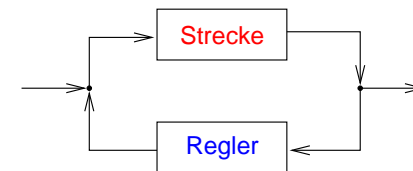
PM = Personen-Monate (Kosten) KDSI = Kilo Delivered Source Instructions (Größe)

Typische Verwendung **empirischer Methoden** gerade im Software Engineering

- Programme und Digitale Systeme sind formale Objekte
 - besitzen exakte mathematische Modelle (denotational oder operational)
 - Realität = Modell (modulo komplexer Semantik, Compilerbugs, Hardwareversagen, ...)
 - **Eigenschaften der Modelle gelten auch für die Realität**
- Eigenschaften von Modellen zu beweisen ist aufwendig
 - für Software i.Allg. **unentscheidbar**
 - für Hardware i.Allg. in **NP** oder **PSPACE**
- gilt nur für **Funktionale Eigenschaften**, nicht für **Quantitative Aspekte**
 - Verfügbarkeit, Durchsatz, Latenzzeit etc. sind schwer exakt zu modellieren

- Gegenstand der Mathematik sind eigentlich keine Modelle
 - die Natürlichen Zahlen sind "einfach da"
Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk [Kronecker]
 - keine Interpretation notwendig: Untersuchungsgegenstand = Realität
- Modellbegriff in der Mathematischen Logik
 - Lässt sich Mathematik durch Mathematik exakt modellieren?
Nein, nicht im Allgemeinen [Gödel]
 - gewisse Formalisierung sind dennoch möglich:
eine Vielzahl von Sätzen läßt sich **formal herleiten** (d.h. durch symbolisches Schließen in einem formalen Kalkül)

Modellierung eines Reglers im Einsatz



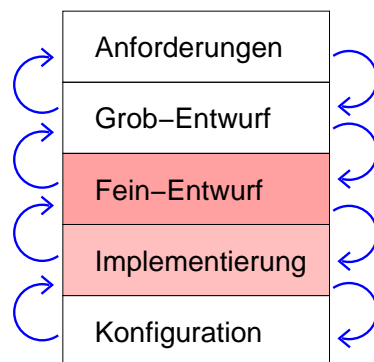
- der Eingriff des Menschen bzw. das technische System wird mitmodelliert
- Modellierung ist Approximation der Realität
 - reales System verhält sich nicht exakt wie das Modell
- Ziel ist Konstruktion und Optimierung des Reglers

- Ziel ist Konstruktion und Optimierung von **Informatiksystemen**
- Modelle zur **quantitativen** Analyse/Optimierung
 - Warteschlangensysteme, Markov-Chains etc.
 - Simulation von Szenarien
- High-Level Modelle
 - schrittweise Verfeinerung/Synthese (z.B. Code-Erzeugung, Compiler)
 - Beispiel: Model Driven Architecture (MDA) = ausführbare UML Modelle
 - Beispiel: Behavioural Models und Synthesis für Digitale Systeme

- Modelle im Sinne der Naturwissenschaft
 - Computational Science, Modellierung mit dem Computer
- Weitere Empirische Modelle
 - empirische Methoden im Software Engineering
- Mathematische Modelliermethodik
 - Logik als Basis, SW/HW als Formel, Realität = Modell
- High-Level Modelle
 - qualitativ (funktional) oder quantitativ, Verfeinerung/Synthese

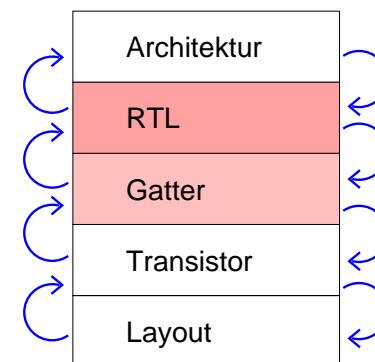
Verfeinerung SW

am Beispiel Software-Entwurf



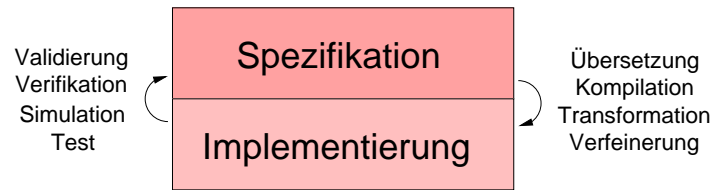
Verfeinerung HW

am Beispiel Hardware-Entwurf



Verifikation

Synthese



formal

oder

adhoc