

Modellierung

Prof. Dr. Sibylle Schwarz
HTWK Leipzig, Fakultät IM
Gustav-Freytag-Str. 42a, 04277 Leipzig
Zimmer ZU 411 (Zuse-Bau)
<https://informatik.htwk-leipzig.de/schwarz>
sibylle.schwarz@htwk-leipzig.de

Wintersemester 2026/27

Informatik

Informatik Lehre von der **symbolischen Darstellung** und **Verarbeitung** von **Information** durch **Algorithmen**

Teilgebiete der Informatik:

theoretisch

- ▶ Sprachen zur Formulierung von Information und Algorithmen,
- ▶ Möglichkeiten und Grenzen der Berechenbarkeit durch Algorithmen,
- ▶ Grundlagen für technische und praktische (und angewandte) Informatik

technisch

- ▶ maschinelle Darstellung von Information
- ▶ Mittel zur Ausführung von Algorithmen

(Rechnerarchitektur, Hardware-Entwurf, Netzwerk, ...)

praktisch Entwurf und Implementierung von Algorithmen
(Betriebssysteme, Compilerbau, SE, ...)

angewandt Anwendung von Algorithmen
(Text- und Bildverarbeitung, Datenbanken, KI, Medizin-, Bio-, Wirtschafts-, Medien-Informatik, ...)

Beispiel Münzenspiel

Spielfeld:

(unendlich viele) Stapel von Münzen nebeneinander



1. Zu Beginn liegen 5 Münzen auf einem Stapel, alle anderen Stapel sind leer.
2. In jedem Zug werden zwei Münzen von einem Stapel (auf dem wenigstens zwei Münzen liegen) genommen und eine davon auf den rechten, die andere auf den linken Nachbarstapel gelegt.
3. Das Spiel ist zuende, wenn kein Zug mehr möglich ist.

Münzenspiel: Fragen

- ▶ In welchen Spielzuständen sind keine Züge möglich?
- ▶ Welche Zustände sind aus dem Startzustand erreichbar?
- ▶ Wieviele Züge können (mindestens / höchstens) gespielt werden, bis kein Zug mehr möglich ist?

Münzenspiel für zwei Personen mit den zusätzlichen Regeln:

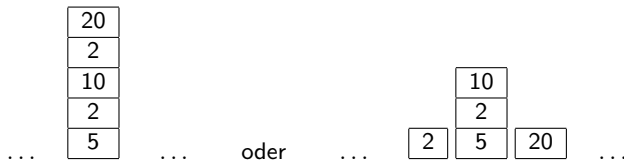
4. Beide Spieler ziehen abwechselnd.
 5. Wer am Zug ist, wenn kein Zug mehr möglich ist, verliert.
(Der andere gewinnt.)
- ▶ Kann der Spieler, der den ersten Zug macht, gewinnen?
 - ▶ Wie kann der Spieler, der den ersten Zug macht, gewinnen?
 - ▶ Gewinnt immer der Spieler, der den ersten Zug macht?
 - ▶ (Wie) Hängt das von der Anzahl der Münzen zu Beginn ab?

Modellierung des Münzenspiels: Zustände



Zustand (Konfiguration): Momentaufnahme des Spielfeldes
vor oder nach einem (vollständigen) Spielzug
formale (maschinenlesbare) Modellierung eines Spielzustandes, z.B. durch

- Folge von Stapeln (Folgen) von Münzen verschiedener Werte, z.B.



- Folge natürlicher Zahlen (Anzahlen der Münzen je Stapel),
z.B. [..., 0, 5, 0, ...] oder [..., 0, 1, 3, 1, 0, ...]
- endliche Folge natürlicher Zahlen (nur relevanter Bereich),
z.B. [5] oder [1, 3, 1] oder [0, 5, 0] oder [0, 1, 3, 1, 0]

Abstraktion von Art und Anzahl der Objekte
ermöglicht einfache Übertragung auf ähnliche Aufgaben

Modellierung des Münzenspiels: Ablauf

Modellierung aller möglichen Übergänge zwischen Zuständen
z.B. als

- ▶ formale Darstellung der Spielregeln als (beidseitig unendliche) Folge mit den Positionen $\mathbb{Z} = [\dots, -2, -1, 0, 1, 2, \dots]$

Startzustand : $[\dots, 0, 5, 0 \dots]$

Spielzug: Ersetzung einer Teilfolge des Zustandes der Form $[x, y + 2, z]$ durch $[x + 1, y, z + 1]$

Regel: $[x, y + 2, z] \rightarrow [x + 1, y, z + 1]$

Spielende , wenn jedes Element der Folge < 2
(kein Zug mehr möglich)

- ▶ formale (graphische) Darstellung aller möglichen Spielabläufe

Modellierung des Verlaufes eines Spieles z.B. als

- ▶ Weg im Spielgraphen
- ▶ (gültige) Folge von Positionen in $\mathbb{Z}$,
an der die (hier einzige) Regel angewendet wurde

Modelle

Modelle sind **Abstraktionen** (Vereinfachungen)
realer Dinge, Eigenschaften, Beziehungen, Vorgänge

- ▶ Auswahl der (für den Anwendungsbereich, zur Problemlösung) wichtigen Informationen
- ▶ Vernachlässigung unwichtiger Details

Beispiele:

- ▶ Spielzustände im Münzspiel:
 - wichtig: lineare Anordnung der Positionen (Spielfeld), Zuordnung von (Münz-)Anzahlen zu Positionen,
 - unwichtig: Art und Werte der Münzen, Abstände der Stapel, ...
- ▶ Liniennetzplan
 - wichtig: Stationen, Verbindungen zwischen Stationen, Art der Verbindung (z.B. Linienummer)
 - unwichtig: genaue Linienführung, aktuelle Verspätungen, Baustellen, ...
- ▶ Grundriss, Stundenplan, Ablaufplan, Kostenplan
- ▶ Holzmodelle von Gebäuden, Modellfahrzeuge
(<https://smart-driving.htwk-leipzig.de/>)

Art der Modelle abhängig von **Problem**bereich und geplanter **Verwendung**

Prozess beim Lösen von Aufgaben (Problemen)

Analyse der (informalen) Aufgabe, Identifikation von

- ▶ Aufgabenbereich (Kontext)
- ▶ Frage (Typ, mögliche Werte, Eigenschaften der Eingabedaten)
- ▶ gewünschte Lösung (Ausgabe: Typ, Eigenschaften, Zusammenhang mit Eingabe)

Modellierung (Abstraktion und formale Darstellung) von

- ▶ Aufgabenbereich (Kontext)
- ▶ Anforderungen an Eingaben
- ▶ Anforderungen an Lösungen

Modellierung von **Daten**

und deren **Eigenschaften** und **Beziehungen** zueinander

Entwurf einer Lösungsstrategie für die modellierte Aufgabe
(mit vorhandenen oder neuen Methoden)

Modellierung von **Abläufen und deren Eigenschaften**

Umsetzung der Lösungsstrategie im Modellbereich

Ausführung der Lösungsstrategie im Modellbereich

Übertragung der Lösung vom Modellbereich in die Realität

Formalisierung: Syntax und Semantik

algorithmische (maschinelle) **Verarbeitung** von Information
erfordert geeignete **Darstellung** der Information

- ▶ Formalismus (Kalkül):
formale Sprache, Maschinen-(und Menschen-)lesbar
- ▶ eindeutige Zuordnung einer Bedeutung zu den Elementen des Formalismus

Semantik (Bedeutung)
Was wird dargestellt?

Syntax (Darstellungsform)
Wie wird es dargestellt?
(meist viele verschiedene Möglichkeiten)

Beispiel: Münzspiel

Formale Methoden

Abstraktion von unwichtigen Details
(Entwicklung und Verwendung von Modellen)

Präzisierung der relevanten Aussagen
(eindeutige Semantik)

Systematisches Lösen (auch maschinell) von formal dargestellten
Problemen möglich

Struktureigenschaften formaler Beschreibungen
Schlussweisen unabhängig von Bedeutung der
Aussagen

Aus der Modulbeschreibung

Modul C114 Modellierung (8 ECTS-Punkte)

Arbeitsaufwand: Präsenzzeit 6 SWS = 4 SWS V + 2 SWS S
Selbststudienzeit 156 h

Lernziele: Die Studierenden können mathematische und logische Grundkonzepte zur Modellierung praktischer Aufgabenstellungen anwenden.
Sie können Anforderungen an Software und Systeme formal beschreiben und wissen, dass deren Korrektheit mit formalen Methoden nachweisbar ist.

Inhalt der Lehrveranstaltung Modellierung

Einführung in formale Beschreibungsverfahren in der Informatik
Modellierung von

Aussagen durch aussagenlogische Formeln

Daten durch

- ▶ Mengen
- ▶ Wörter (Folgen) und Sprachen
- ▶ Terme

Zusammenhängen (Beziehungen) durch

- ▶ Relationen
- ▶ Graphen
- ▶ Strukturen
- ▶ Abstrakte Datentypen

Abläufen durch Zustandübergangssysteme

Eigenschaften, Anforderungen (für Daten und Systeme) durch
Formeln der Prädikatenlogik (der ersten Stufe)
(Ausblick in nichtklassische Logiken)

Lernziele (und Nebenwirkungen)

- ▶ Fähigkeit zur Abstraktion
- ▶ Verständnis der grundlegenden Modellierungs-Formalismen der Informatik:
Logik, Mengen, Relationen, Graphen, Terme, Strukturen, Datentypen
- ▶ anwendungsbereite Kenntnisse zur Modellbildung
- ▶ Zusammenhänge zu anderen Gebieten der Informatik und zur Mathematik

Literatur

Folien zur Vorlesung, jeweils nach der Vorlesung veröffentlicht

[https:](https://informatik.htwk-leipzig.de/schwarz/lehre/ws26/modellierung)

[//informatik.htwk-leipzig.de/schwarz/lehre/ws26/modellierung](https://informatik.htwk-leipzig.de/schwarz/lehre/ws26/modellierung)

empfohlene Bücher:

- ▶ zur Modellierung:
 - ▶ Uwe Kastens, Hans Kleine Büning: Modellierung - Grundlagen und formale Methoden, Hanser 2008
- ▶ zur Logik
 - ▶ Michael Huth, Mark Ryan: Logic in Computer Science, Cambridge University Press 2010
 - ▶ Uwe Schöning: Logik für Informatiker, Spektrum, 1995
 - ▶ Martin Kreuzer, Stefan Kühling: Logik für Informatiker, Pearson Studium, 2006

Organisation der Lehrveranstaltung

Folien, Übungsserien, Termine, Änderungen, ... unter <https://informatik.htwk-leipzig.de/schwarz/lehre/ws26/modellierung>

Präsenzstudium

6 SWS

Lehrveranstaltungen für jeden Studenten:

Vorlesung jeden Donnerstag und Freitag (4SWS)

Seminar (5 Gruppen) Montag bis Mittwoch (2SWS)

Gäste, Wiederholer (vor JG 26): mit **MIB**-Gruppen

- ▶ Besprechung der Übungsserien (Vorrechnen)
- ▶ Fragen zum aktuellen Vorlesungsinhalt

Selbststudium: (incl. PV) **156 h $\approx$ 10 h/Woche**

Vor- und Nachbereitung der Vorlesungen anhand

Folien und angegebener Literatur

Übungsaufgaben zu jeder Vorlesung (wöchentlich einzusenden)

klar, überschaubar, Feedback

schriftliche Übungsserien (vor Donnerstag)

praktische Aufgaben im Autotool (vor Montag)

Literaturstudium ergänzend, Beitrag zur Lösung der Aufgaben

Selbststudium

Vor- und Nachbereitung **jeder** Lehrveranstaltung

(Vorlesung, Seminar, Praktikum, ...)

Unterlagen zu den Lehrveranstaltungen (Folien, eigene Notizen)

durcharbeiten

(enthalten auch zum Lösen der Übungsaufgaben notwendige Definitionen, Herleitungen, Beispiele, ...)

Übungsaufgaben **regelmäßig** und **rechtzeitig** lösen,

(Aufgaben vom Typ der) Übungsaufgaben gehören zum Inhalt des Moduls und werden geprüft

Fachliteratur benutzen (z.B. Bücher, E-Books in Bibliothek),
enthalten Erklärungen, zusätzliche Übungsaufgaben
(Internet-Quellen sind oft unzuverlässig)

Lerngruppen bilden und **gemeinsam** lernen und Aufgaben lösen

Nachfragen bei Dozenten (E-Mail, Sprechzeit, nach der Lehrveranstaltung, ...), Mitstudenten, älteren Studenten, Fachschaft, ...

Schriftliche Übungsaufgaben (Übungsserien)

- ▶ gestellt jeweils am Donnerstag
(Aufgaben zu den Vorlesungen am Donnerstag und Freitag)
- ▶ Lösungen bis zum folgenden Mittwoch ausschließlich online über OPAL einzusenden (wie in der Einführungswoche erklärt)
genau **eine pdf-Datei je Aufgabe**, Dateiname serieX-aufgY-Z.pdf
für Lösung zu Aufgabe Y von Übungsserie X vom den Studierenden mit Familiennamen Z,
- ▶ Lösung in Gruppen aus ≤ 3 Studierenden zulässig (und empfohlen),
Namen **aller Mitglieder der Gruppe** oben auf der Lösung vermerken,
jedes Gruppenmitglied sendet die (gemeinsame) Lösung zu Opal
Dateiname serieX-aufgY-Z1-Z2-Z3.pdf
- ▶ Bewertung bis zum folgenden Sonntag
(0: falsch, 1: überwiegend richtig, 2: korrekt)
- ▶ Besprechung in der darauffolgenden Übung
Vorträge (Vorrechnen) zu den Aufgabenlösungen

Übungsserien auch unter <https://informatik.htwk-leipzig.de/schwarz/lehre/ws26/modellierung>

Nicht im Opal-Kurs eingesendete Lösungen werden **nicht gewertet**.

Seminar (Übungen)

Lernziele bei der Bearbeitung der Übungsaufgaben:

- ▶ Nachbereitung der letzten Vorlesung anhand der Vorlesungsfolien und Literatur (z.B. der angegebenen)
- ▶ Vorbereitung der nächsten Vorlesung
- ▶ Vorbereitung der Seminarvorträge zu jeder Aufgabe

Seminar (Übungen):

- ▶ Besprechung der Lösungen der schriftlichen Übungsaufgaben, Vorrechnen zur Prüfungszulassung
- ▶ Fragen zum aktuellen Vorlesungsstoff und zu den neuen schriftlichen und praktischen Übungsaufgaben
- ▶ **in jede Übung mitbringen:** (alles sofort les- und auffindbar) alle bisherigen Vorlesungsfolien und eigene Notizen dazu, Übungsseries und eigene Lösungen

Praktische Hausaufgaben – Autotool

<https://autotool.imn.htwk-leipzig.de/new>

(Einführung am vergangenen Donnerstag)

einmal zu Beginn:

- ▶ Anmeldung über Shibboleth (HRZ-Login)
- ▶ Click „Einschreibung“,
eigene Übungsgruppe auswählen

jedes Mal:

- ▶ Anmeldung über Shibboleth (HRZ-Login)
- ▶ Aufgabe ansehen (Click „Bearbeiten“)
- ▶ Aufgabe lösen (meist mit Stift, Papier, Folien, Literatur),
Lösung in Textfeld eintragen,
Click „Textfeld absenden“
- ▶ Autotool-Antwort lesen und verstehen !

Prüfung

Zulassungsvoraussetzungen (Prüfungsvorleistungen):
regelmäßige erfolgreiche Lösung der Übungsaufgaben,
also **alle** folgenden drei Bedingungen:

- ▶ rechtzeitige Einsendung von richtigen Lösungen
(mit wenigstens 1 Punkt bewertet) zu wenigstens 70% aller
gestellten schriftlichen Übungsaufgaben (OPAL)
- ▶ Präsentation: mindestens drei Vorträge in den Seminaren
(richtiges „Vorrechnen“ der Lösungen)
je einmal bis Ende November, Dezember und Januar
- ▶ 50% aller Punkte für praktische Pflichtaufgaben (Autotool)

Prüfung: Klausur 120 min

Aufgabentypen aus den Übungsserien

einziges zugelassenes Hilfsmittel:

ein beidseitig handbeschriebenes A4-Blatt

Aussagenlogik

Aussagen

Aussage = Behauptung

Beispiele:

- ▶ Es regnet.
- ▶ Die Straße ist nass.
- ▶ 9 ist eine Primzahl.
- ▶ $\sqrt{2} \in \mathbb{Q}$
- ▶ $3 < 5$
- ▶ $x < 5$ (hängt von x ab, keine Aussage)
- ▶ Ist $x < 5$? (keine Aussage)
- ▶ Sei $x < 5$. (keine Aussage)
- ▶ Morgen regnet es.
- ▶ Es ist nicht alles Gold, was glänzt.

Wahrheitswerte

Prinzipien der **klassischen** Logik:

Zweiwertigkeit: Jede Aussage ist wahr oder falsch.

ausgeschlossener Widerspruch:

Keine Aussage ist sowohl wahr als auch falsch.

Wahrheitswerte 1 (wahr) oder 0 (falsch)

Jede Aussage p hat genau einen Wahrheitswert $W(p) \in \{0, 1\}$.

Beispiele:

- ▶ $W(\text{Es regnet.}) = ?$
- ▶ $W(\text{Die Straße ist nass.}) = ?$
- ▶ $W(9 \text{ ist eine Primzahl.}) = 0$
- ▶ $W(\sqrt{2} \in \mathbb{Q}) = 0$
- ▶ $W(3 < 5) = 1$
- ▶ $W(\text{Morgen regnet es.}) = ?$
- ▶ $W(\text{Es ist nicht alles Gold, was glänzt.}) = 1$

Zusammengesetzte Aussagen – Junktoren

Junktor (mit zugeordneter Stelligkeit):

Symbol (Syntax) für **Verknüpfung** von Aussagen

z.B. „und“ (zweistellig), „nicht“ (einstellig)

Gottlob Frege (1848–1925):

Die Bedeutung des Ganzen ist eine Funktion der Bedeutung seiner Teile.

Der Wahrheitswert einer zusammengesetzten Aussage lässt sich aus den Wahrheitswerten ihrer Teilaussagen berechnen.

Semantik (Bedeutung) eines n -stelligen Junktors $*$:

$\llbracket * \rrbracket : \{0, 1\}^n \longrightarrow \{0, 1\}$

(n -stellige Funktion auf der Menge $\{0, 1\}$)

Wahrheitswertkonstanten (nullstellige Junktoren):

$\mathfrak{t}$ mit $\llbracket \mathfrak{t} \rrbracket = 1$

$\mathfrak{f}$ mit $\llbracket \mathfrak{f} \rrbracket = 0$

Konjunktion $\wedge$

Es regnet **und** 9 ist eine Primzahl.

- ▶ $W(9 \text{ ist eine Primzahl.}) = 0$
- ▶ $W(\text{Es regnet.}) = ?$
- ▶ $W(\text{Es regnet und } 9 \text{ ist eine Primzahl.}) = 0$

$p \wedge q$ ist genau dann wahr,
wenn **beide Aussagen p und q** wahr sind.

$W(p)$	$W(q)$	$W(p \wedge q)$
0	0	0
0	1	0
1	0	0
1	1	1

$$W(p \wedge q) = \min(W(p), W(q))$$

$\llbracket \wedge \rrbracket = \min$ ist kommutativ, assoziativ

$$\bigwedge_{i=1}^n p_i = p_1 \wedge p_2 \wedge \cdots \wedge p_n$$

Disjunktion $\vee$ (inklusive)

Es regnet oder $3 < 5$.

- ▶ $W(3 < 5) = 1$
- ▶ $W(\text{Es regnet}) = ?$
- ▶ $W(\text{Es regnet oder } 3 < 5.) = 1$

$p \vee q$ ist genau dann wahr,
wenn **wenigstens eine der Aussagen p und q** wahr ist.

$W(p)$	$W(q)$	$W(p \vee q)$
0	0	0
0	1	1
1	0	1
1	1	1

$$W(p \vee q) = \max(W(p), W(q))$$

$\llbracket \vee \rrbracket = \max$ ist kommutativ, assoziativ

$$\bigvee_{i=1}^n p_i = p_1 \vee p_2 \vee \cdots \vee p_n$$

Negation $\neg$

$\neg(\sqrt{2} \in \mathbb{Q})$ (oft auch $\sqrt{2} \notin \mathbb{Q}$)

► $W(\sqrt{2} \in \mathbb{Q}) = 0$

► $W(\neg(\sqrt{2} \in \mathbb{Q})) = 1$

$\neg p$ ist genau dann wahr, wenn p falsch ist.

$W(p)$	$W(\neg p)$
0	1
1	0

$$W(\neg p) = 1 - W(p)$$

Implikation $\rightarrow$

Wenn es regnet, dann ist die Straße nass.

- ▶ $W(\text{Es regnet.})=?$
- ▶ $W(\text{Die Straße ist nass.})=?$
- ▶ $W(\text{Wenn es regnet, dann ist die Straße nass.})=1$

$p \rightarrow q$ ist genau dann wahr,
wenn die Aussage p falsch oder die Aussage q wahr ist.

$W(p)$	$W(q)$	$W(p \rightarrow q)$
0	0	1
0	1	1
1	0	0
1	1	1

$$W(p \rightarrow q) = \begin{cases} 1 & \text{falls } W(p) \leq W(q) \\ 0 & \text{sonst} \end{cases}$$

Äquivalenz $\leftrightarrow$

$3 < 5$ gilt **genau dann, wenn** $0 < 5 - 3$ gilt.

► $W(3 < 5) = 1$

► $W(0 < 5 - 3) = 1$

► $W(3 < 5 \text{ gilt genau dann, wenn } 0 < 5 - 3 \text{ gilt.}) = 1$

$p \leftrightarrow q$ ist genau dann wahr, wenn
entweder **beide Aussagen p und q gelten**
oder **beide nicht gelten.**

$W(p)$	$W(q)$	$W(p \leftrightarrow q)$
0	0	1
0	1	0
1	0	0
1	1	1

$$W(p \leftrightarrow q) = \begin{cases} 1 & \text{falls } W(p) = W(q) \\ 0 & \text{sonst} \end{cases}$$