

Julian Kunkel

Informatik, mehr als Programmierung und Digitalisierung

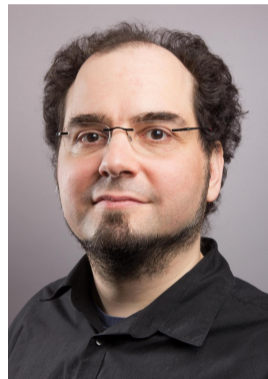


Agenda

- 1 Über mich
- 2 Hochleistungsrechnen
- 3 Probevorlesung: Einführung in Algorithmen
- 4 Informatik

Über Mich – Prof. Dr. Julian Kunkel

- Stellvertretender Leiter GWDG – Bereich Hochleistungsrechnen
 - ▶ Die GWDG ist das Rechenzentrum der Universität und Max-Planck Gesellschaft
 - ▶ Gruppenleiter AG-C
- Professor am Institut für Informatik
- Forschungsfragen (vereinfacht)
 - ▶ Wie können wir größte Datenmengen effizient verarbeiten
 - ▶ Wie können wir 1000 Computer für Wissenschaftliche Fragestellungen nutzen



Mein Weg zur Informatik

- Vater nutze IBM PC für seine Arbeit mit monochrom Bildschirm
 - ▶ Schreibmaschine als "Drucker" angeschlossen
 - ▶ Später MSDos, Windows 3.0/3.1 und richtiger Bildschirm
- Eigener Computer mit 8 Jahren: C64 mit Floppy Laufwerk
 - ▶ C64 hat eine BASIC Version als Betriebssystem
 - ▶ Um ein Program/Spiel zu starten eingeben: Load '*', 8,1

A screenshot of the Commodore 64 BASIC V2 prompt screen. The text is displayed in a monochrome font on a dark background. The text reads: "**** COMMODORE 64 BASIC V2 ****", "64K RAM SYSTEM 38911 BASIC BYTES FREE", "READY", "LOAD '*', 8", "SEARCHING FOR \$", "LOADING", "READY".

```
**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY
LOAD '*', 8
SEARCHING FOR $
LOADING
READY
```

- Nintendo Entertainment System – später eigener PC mit 90 MHz
- Betriebspraktikum bei einer IT-Firma (diese baute u.a. Computer zusammen)
 - Linux 1998 gelernt, dann mit Mitschülern Schulfirewall betreut

Programmiererfahrung

- (etwas) mit C64 Basic gespielt
- (etwas) GW BASIC in Realschule Kunst, Musik,
- (etwas) Turbo Pascal in Gymnasium
- Mit 15 Jahren erste grafische Anwendungen mit Visual Basic programmiert
- Richtig programmieren erst im Studium gelernt!

Warum habe ich mich für Informatik entschieden?

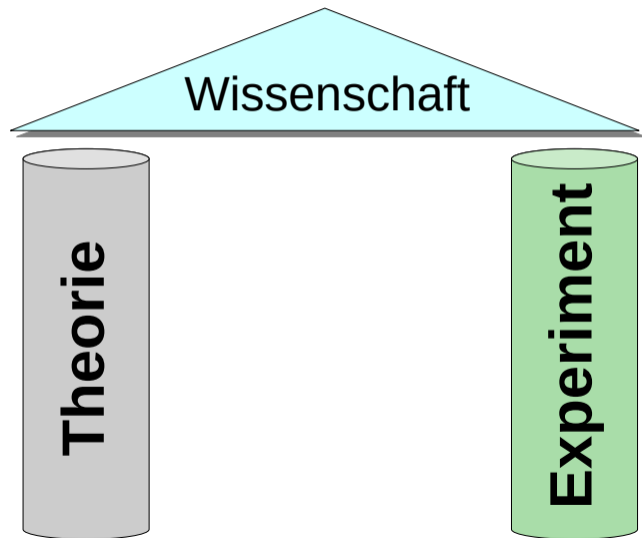
- Ich war interessiert an den Naturwissenschaften
 - Computertechnik lässt sich vielfältig einsetzen
 - ▶ Naturwissenschaften, Ingenieurwissenschaften, Medizin, Kunst, Musik, ...
 - Automatisierung schreitet unaufhörlich voran
 - Berufe in der IT sind ungefährlich und man kann von Zuhause arbeiten
 - Softwareprototypen können verändert werden - kein Einwegeprodukt
 - Ich war nicht gut in Rechtschreibung und Fremdsprachen
 - SciFiction wie Holodeck und Universal Translator fand ich spannend
 - Wiederkehrende Arbeiten langweilen mich
- Ich möchte alles (für mich) optimieren, das erfordert Verständnis

Hochleistungsrechnen

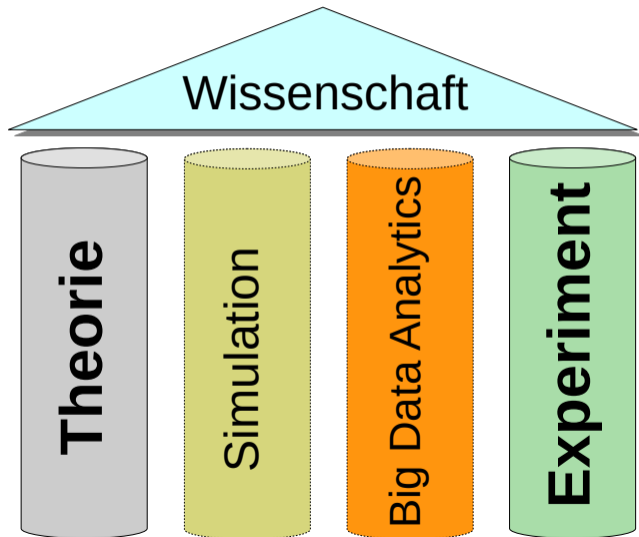
Outline

- 2 Wissenschaftlicher Erkenntnisgewinn
- 3 Hochleistungsrechnen
- 4 Forschung
- 5 KI in Göttingen

Säulen des Erkenntnisgewinns



Säule des Erkenntnisgewinns: **Moderne Ansicht**



Computergestützte Simulation

Modellierung und Simulation der Welt ersetzen das traditionelle Experiment

Computersimulation als Instrument ermöglicht

- Beliebige zeitliche, räumliche Auflösung
- Manipulation beliebiger Parameter
- Reproduzierbar
- Keine Einschränkung durch Kosten, Risiken, Ethik
 - ▶ Was passiert, wenn Atomkraftwerk X explodiert?
 - ▶ Einfluss von Chemikalien auf Menschen
 - ▶ Neuronen-Aktivität in Gehirnzellen
- Vorhersage von künftigen Ereignissen
 - ▶ Wettervorhersage, Klima-Veränderungen
 - ▶ Infektionsgeschehen ...

Simulation ist Rechen- und Speicherintensiv

Beispiel

- Billionen Neuronen des Gehirns
- Flugzeugtriebwerk Modelle aus Billionen von "Elementen"
- KI-Modelle rechnen auf 1000 von Grafikkarten
- Wettersimulation von morgen muss heute fertig werden (Zeitlimit)
 - ▶ Aber auch eine hohe Genauigkeit erreichen, d.h., hohe Auflösung

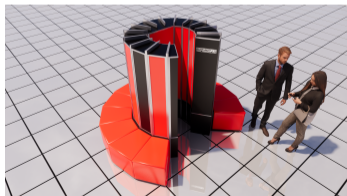
Bewältigung der Anforderungen an Speicher und Rechenleistung

- Einzelne PC oder Server \Rightarrow nicht ausreichend!
- Wir benötigen mehr Leistung ... Hohe Leistung ... **Hochleistungsrechnen**

Hochleistungsrechnen (High-Performance Computing)

Definitionen

- HPC: Feld das ressourcenintensive Rechenprobleme löst
 - ▶ Aufgabe erfordert zu viel Speicher/Rechenzeit für "standard" Computer
 - ⇒ HPC ermöglicht wissenschaftliche Simulation und Erkenntnis
- Supercomputer: Vereinigt Leistung vieler einzelner (unabhängiger) Server
 - ▶ Früher gigantische monolithische Computer
 - ▶ Heutzutage 10-1000 Server zu Cluster verbunden



Nr. 1 Supercomputer der Welt (2025)

El Capitan im Lawrence Livermore National Laboratory, Kalifornien, USA

- Rechenressourcen: 11.136 Knoten
11 Millionen Kerne
- Peakleistung: 2.746.376 Teraflops
 - ▶ Eine PS5 Pro hat 16,7 Teraflops
- Arbeitsspeicher: 9,2 Petabyte
- Storage: 5 Petabyte
- Leistungsaufnahme: 30 Megawatt
- Kosten: ca. 600 Million US\$



Top500 Liste der Welt leistungsstärksten Supercomputer

Hochleistungsrechnen in Göttingen

GWDC: Hochschulrechenzentrum und IT-Infrastruktur

- HPC-Systeme für lokale Wissenschaftler, Deutschland und für DLR
- Integriert Forschung für HPC-Systeme mit Service und Praxis
- GWDC Supercomputer EMMY & KI-Servicezentrum für Deutschland



Herausforderungen des Hochleistungsrechnens

HPC Systeme und Programmierung ist komplex

⇒ Hochleistungsrechnen benötigt Forschung- und Entwicklungsarbeiten

Herausforderungen und Forschungsaufgaben im HPC

- Parallele Programmierung von Anwendungen und Workflows
d.h. gleichzeitige Ausführung von Code **einer** Anwendung
 - ▶ Modelle, Laufzeitumgebungen und Softwareentwicklungsmethodiken
- Effiziente Systemnutzung
 - ▶ Rechenressourcen, Speichersysteme, Rechenzentrum-Infrastruktur
- Lehre und Training für Benutzende und Admins
- ... vieles mehr ...

Künstliche Intelligenz in Göttingen

- KI ist ein zentraler Bestandteil im Informatikstudium in Göttingen, z.B.:
 - ▶ KI-Methoden im akademischen Alltag
 - ▶ Maschinelles Lernen und Deep Learning
 - ▶ Computer Vision und Verarbeitung natürlicher Sprache
 - ▶ KI-gestützte Datenanalyse
- Göttingen bietet ein breites Ökosystem für KI-Forschung und Anwendung

KI-Dienste an der Universität Göttingen

- Dienste für Studierende, Mitarbeitende
 - ▶ **Chat AI** – Textgenerierung
 - ▶ **Image AI** – Bildgenerierung
 - ▶ **Voice AI** – Transkription und Übersetzung
 - ▶ **Protein AI** – Proteinvorhersage
 - ▶ ...

Algorithmen: Probevorlesung

Outline

6 Einführung

7 Structured Program Theorem

8 Beschreibung von Algorithmen

9 KI und Algorithmen

10 Zusammenfassung

Einführung in Algorithmen

- Im Folgenden eine kurze Probevorlesung zur „Einführung von Algorithmen“
 - ▶ Einige Komponenten zum aktiven Lernen sind enthalten
 - ▶ Bitte machen Sie mit!
- Lernziele (Als Teilnehmende sollten Sie in der Lage sein:)
 - ▶ Beispiele für die Relevanz von Algorithmen im täglichen Leben zu geben
 - ▶ Die Eigenschaften von Algorithmen zu definieren
 - ▶ Das Structured Program Theorem wiederzugeben

Algorithmus

Definition

Ein Algorithmus ist eine **eindeutige Handlungsvorschrift** zur **Lösung** eines Problems. Algorithmen bestehen aus **endlich vielen, wohldefinierten Einzelschritten**.

Quelle: Wikipedia¹

Algorithmen sind in allen Teilgebieten der Informatik relevant:

Algorithms are the threads that tie together most of the subfields of computer science.

Donald Knuth

Eigenschaften von Algorithmen

Verarbeitung



- **Determiniertheit:** bei jeder Ausführung mit gleichen Eingaben wird die gleiche Ausgabe generiert.
- **Terminiertheit:** Ein Algorithmus terminiert, wenn er nach endlich vielen Schritten anhält - für jede mögliche Eingabe.
- **Effektivität:** Der Effekt jeder Anweisung eines Algorithmus muss eindeutig festgelegt sein.

Aktives Lernen: Fragezeit

Überlegen Sie einen Moment für sich selbst

- Können Sie ein Beispiel für einen Algorithmus geben?
- Haben Sie einen Algorithmus schon einmal ausgeführt?
- Zeit: 2 Minuten

Definition

Ein Algorithmus ist eine **eindeutige Handlungsvorschrift** zur **Lösung** eines Problems. Algorithmen bestehen aus **endlich vielen, wohldefinierten Einzelschritten**.

Quelle: Wikipedia²

Beispielalgorithmen

- Algorithmen sind überall
- Tagtägliche Aktivitäten
 - ▶ Kochrezepte befolgen
 - ▶ Duschen, Kaffee machen, ein Auto fahren
 - ▶ Bearbeitung von Supermarkt Warteschlange
- Eingebettete Systeme
 - ▶ Liftsteuerung
 - ▶ Geldautomat
 - ▶ Biometrische Identifikation am Smartphone
- Mathematik
 - ▶ Gegeben zwei Zahlen, identifiziere die größere
 - ▶ Den Umfang eines Kreises berechnen

Weitere Eigenschaften von Algorithmen

- **Korrektheit:** Gültige Eingaben \Rightarrow korrekte Ausgabe
- **Portabilität:** Algorithmus funktioniert unabhängig von
 - ▶ der "Hardware" die den Algorithmus ausführt (oder durch den Menschen!)
 - ▶ der Notation der Algorithmusbeschreibung (Programmiersprache)
- **Effizient**
 - ▶ Sparsamer Umgang von Rechenressourcen (Speicherbedarf)
 - ▶ Kurze Laufzeit (Zeitkomplexität)

Outline

- 6 Einführung
- 7 Structured Program Theorem**
- 8 Beschreibung von Algorithmen
- 9 KI und Algorithmen
- 10 Zusammenfassung

Structured Program Theorem (Böhm-Jacopini Theorem)

Jedweder Algorithmus kann aus elementaren Operationen mittels der folgenden Kontrollstrukturen aufgebaut werden:

- **Sequenz:** Operationen nacheinander ausführen (erst Op1, dann Op2, ...)
- **Selektion:** Basierend auf dem Ergebnis eines Ausdrucks wird eine von zwei Operationen ausgeführt
- **Iteration:** Wiederholung einer Operation so lange wie ein Ausdruck wahr ist

Hierbei bauen wir mittels der Konstruktstrukturen komplexere Operationen auf.

D.h. wir können ALLE Algorithmen ausführen/beschreiben, wenn wir diese Konstrukte verstehen!

Sequenz

- Führt einen Operation nach der nächsten aus
- Alle Operationen werden in einer Sequenz ausgeführt
 - ▶ Zuerst tue dies
 - ▶ Dann tue das
 - ▶ Dannach das...
- Es ist sinnvoll die Schritte durchnummerieren
- Beispielproblem: Ich bin durstig
 - 1 Öffne den Kühlschrank
 - 2 Nimm den Saft heraus
 - 3 Trinke den Saft
- Der Entwickler eines Algorithmus definiert die Reihenfolge
 - ▶ Im Beispiel würde keine andere Reihenfolge sinnvoll funktionieren

Selektion/Auswahl

- Selektion führt eine Operation basierend auf einer Bedingung aus
 - ▶ Typischerweise: Ja/Nein, Wahr/Falsch Entscheidungen
- Alternative Darstellung
 - ▶ Falls das wahr ist, dann tue X
 - ▶ Andernfalls, mache Y (oder tue nichts)
- Beispiele
 - ▶ Falls es regnet, nimm einen Regenschirm mit
 - ▶ Falls es sonnig ist, benutze Sonnencreme für dein Gesicht
- In Programmiersprachen spricht man oft von bedingten Anweisungen/Verzweigungen

Iteration/Wiederholung

- Iteration kommt von dem Wort "iterieren", d.h. wiederholen
- Iteration wiederholt eine Operation
- Es gibt zwei Grundstrukturen
 - ▶ Explizite Anzahl von Iterationen
 - Beispiel: wiederhole das 10x
 - Beispiel 2: Für jede Person im Raum, gib ihr ein Faltblatt
 - ▶ Implizite Anzahl von Iterationen
 - Wiederhole etwas so lange bis eine Bedingung zutrifft
 - Beispiel: So lange es sonnig ist, spiele im Garten

Outline

- 6 Einführung
- 7 Structured Program Theorem
- 8 Beschreibung von Algorithmen**
- 9 KI und Algorithmen
- 10 Zusammenfassung

Beschreibung von Algorithmen

- Beschreibungen sind essentiell um Algorithmen zu teilen oder zu verstehen
 - ▶ Dank dem Program Theorem müssen wir nur die Konstrukte beschreiben!
- Es gibt viele Möglichkeiten einen Algorithmus zu beschreiben
 - ▶ Textform
 - Natürliche Sprache
 - Pseudocode
 - Programmiersprache
 - ▶ Graphische Repräsentationen
- Die Eigenschaften von einem Algorithmus müssen erhalten bleiben
 - ▶ Effektivität, Determiniertheit, Terminiertheit
- Lesbarkeit/Verständlichkeit ist sehr wichtig
- Standardisierte Repräsentationen gestatten uns leicht Algorithmen zu erlernen/dokumentieren – Standards sind wichtig!

Natürliche Sprache

■ Beispiel: Lasagnerezept

Zutaten für 4 Portionen:

	Olivenöl
500g	Hackfleisch, gemischtes
1	Zwiebel(n)
2	Knoblauchzehe(n)
1 Bund	Petersilie oder TK
1 EL	Tomatenmark
1 Dose	Tomate(n), geschälte (800 g)
etwas	Rotwein

Zubereitung:

In einem Topf das Olivenöl erhitzen, das Hackfleisch darin rundherum anbraten und die gehackten Zwiebeln und die gehackte Petersilie dazugeben. Knoblauch in feinen Scheiben und Tomatenmark dazu rühren und mitbraten. Mit den Dosentomaten aufgießen, salzen und pfeffern. Rotwein nach Belieben beifügen. Die Bolognese mindestens eine halbe Stunde lang bei geöffnetem Topf einkochen lassen.

In einer gebutterten, feuerfesten Form etwas Bolognese verteilen, eine Schicht Lasagneplatten darauf legen, die Nudelschicht wieder mit Bolognese und dann mit einer Schicht Béchamel bedecken. Anschließend wieder eine Schicht Nudeln, Bolognese und Béchamel. So Schicht für Schicht die Form füllen. Die letzte Schicht sollte die Béchamelsauce bilden. Dick mit geriebenem Käse bestreuen und Butterflöckchen darauf setzen.

Die Lasagne im heißen Backofen bei 180 °C Umluft ca. 30 - 40 Minuten backen, bis die Kruste goldbraun ist.

Frage: Ist das eine gute Beschreibung für einen Algorithmus?

- **Determiniertheit:** mit gleichen Eingaben wird die gleiche Ausgabe generiert.
- **Terminiertheit:** der Algorithmus terminiert
- **Effektivität:** der Effekt jeder Operation ist eindeutig festgelegt

Wie verständlich ist der Algorithmus für Sie?

Aktives Lernen: Diskussion des Algorithmus

- Frage: Ist das eine gute Beschreibung für einen Algorithmus?
- Werden unsere Eigenschaften an einen Algorithmus erfüllt:
 - ▶ **Determiniertheit:** mit gleichen Eingaben wird die gleiche Ausgabe generiert.
 - ▶ **Terminiertheit:** der Algorithmus terminiert
 - ▶ **Effektivität:** der Effekt jeder Operation ist eindeutig festgelegt
- Wie verständlich ist der Algorithmus für Sie?

Diskussion

Zutaten für 4 Portionen **[gut zu wissen für wieviele, d.h. Performance]**

In einem Topf das **[wievieviel?]** Olivenöl erhitzen, das Hackfleisch darin rundherum anbraten **[was bedeutet rundherum?]** und die gehackten Zwiebeln und die gehackte Petersilie dazugeben. Knoblauch in feinen Scheiben und Tomatenmark dazu rühren und mitbraten **[wie lange?]**. Mit den Dosentomaten aufgießen, salzen und pfeffern **[wie viel?]**. Rotwein nach Belieben beifügen. Die Bolognese mindestens eine halbe Stunde lang bei geöffnetem Topf einkochen lassen. **[woran merke ich, dass sie fertig ist?]** Die Lasagne im heißen Backofen bei 180 °C Umluft ca. 30 - 40 Minuten backen, bis die Kruste goldbraun ist. **[Wie lange dauert es das Rezept zuzubereiten? Wäre gut zu wissen!]**

Pseudocode

- Pseudocode ist etwas formalisiertere Repräsentation eines Algorithmus
- Pseudo bedeutet etwas "vorgeben"
- Pseudocode gibt vor in einer Programmiersprache verfasst zu sein
 - ▶ Nutzt hierbei Namen für Programmiersprachenkonstrukte
 - ▶ Die Syntax von Pseudocode selbst ist nicht standardisiert
- Beispiel: Problem - wie starte ich ein Auto

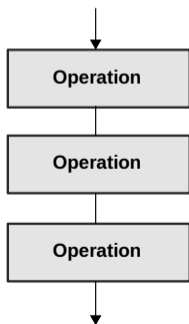
```
1   Stecke den Schluessel ins Zuendschloss
2   Drehe den Schluessel in die Startposition
3
4   Solange das Auto nicht startet:
5       Halte den Schluessel in Startposition
```

- Dieser "Algorithmus" erfüllt übrigens nicht all unsere Kriterien...

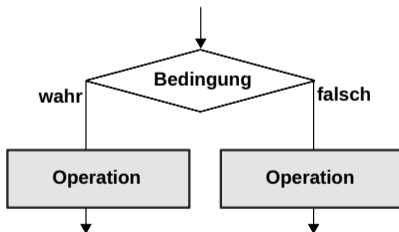
Graphische Repräsentation mittels Flowcharts

- Flowcharts sind standardisierte graphische Repräsentationen der Konstrukte
- Es gibt diverse graphische Notationen/Programmiersprachen

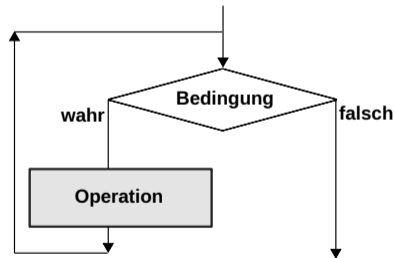
Sequenz



Selektion



Iteration



Algorithmen und Bezug zu KI



- KI erlernt von Ein-/Ausgabedaten die Verarbeitungsschritte den Algorithmus
 - ▶ KI ist typischerweise datengetrieben, lernt Muster und Statistiken
 - ▶ Erlernete Verarbeitung ermöglicht Approximation/Annäherung an Daten
- Zum “Lernen” werden deterministische Algorithmen eingesetzt

Herausforderungen für KI

- Fehlerhafte Zusammenhänge zwischen Ein-/Ausgabe werden erlernt
- Mangelnde Ein-/Ausgabefälle (seltene Fälle) in Daten
- Inkonsistente (widersprüchliche) Daten
Beobachte: X, Ergebnis: Sonne vs. Beobachte: X, Ergebnis: Regen
- Annäherung ist fehlerbehaftet - nicht perfekt
- Verständnis wie das Ergebnis zustande kommt

Herausforderungen für KI

- Fehlerhafte Zusammenhänge zwischen Ein-/Ausgabe werden erlernt
- Mangelnde Ein-/Ausgabefälle (seltene Fälle) in Daten
- Inkonsistente (widersprüchliche) Daten
Beobachte: X, Ergebnis: Sonne vs. Beobachte: X, Ergebnis: Regen
- Annäherung ist fehlerbehaftet - nicht perfekt
- Verständnis wie das Ergebnis zustande kommt

Vergleich zum Menschen

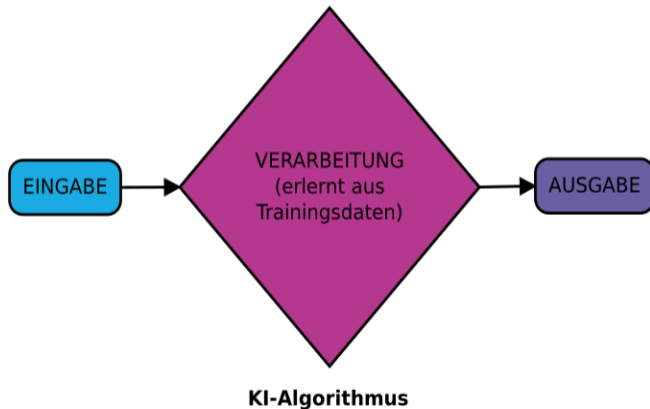
- Der Mensch lernt ähnlich - und verallgemeinert bspw.
- Auch hier führen die Herausforderungen zu den selben Fehlern!
- Fehlerquellen im Deep Learning gelten auch für Menschen

Zusammenfassung

■ Algorithmus

- ▶ Eindeutige Handlungsvorschrift zur Problemlösung
- ▶ Besteht aus Sequenz, Selektion, Iteration
- ▶ Wichtige Eigenschaften: Determiniertheit, Terminiertheit, Effektivität
- ▶ KI nutzt Algorithmen, um aus Daten zu lernen
 - Fehlerhafte Trainingsdaten führen zu fehlerhaften Ergebnissen
 - KI ist ein Werkzeug, ersetzt aber keine Informatikexpert:innen

KI-Algorithmus



Ausblick

- Beschreibung von Algorithmen in Sprache und Code
- Programmiersprachen ermöglichen ausführbare Algorithmen

Informatik

Informatik

Definition: Informatik

Wissenschaft von der systematischen Verarbeitung von Informationen, insbesondere mithilfe von Computern

- Setzt sich aus dem Wort Information und Automatik zusammen.
- Ursprünge in der Mathematik, Nachrichtentechnik und Elektrotechnik

Teilbereiche (Stark vereinfacht)

- Theoretische Info. - Abstraktion/grundlegendes der Info.verarbeitung
- Praktische Informatik - Anwendung auf Probleme der realen Welt
- Technische Informatik - Entwurf/Umsetzung von Hardware/systemnah
- Angewandte Informatik - Zusammenhang mit Anwendungsgebiet
- Interdisziplinäre Wissenschaften
- Künstliche Intelligenz - Macht Maschinen intelligenter
- Informatik und Gesellschaft - Rolle in der Informationsgesellschaft

Angewandte Informatik (B.Sc.) in Göttingen³

- Regelstudienzeit: 6 Semester
- Professor:innen: 19
- Stärke in Anwendung und KI/Datengetriebenen Wissenschaften
- Keine technische Informatik

Exemplarischer Studienverlauf

Semester 1	Semester 2	Semester 3	Semester 4	Semester 5	Semester 6
Mathematik für Studierende der Informatik I	Mathematik für Studierende der Informatik II	Diskrete Stochastik	Theoretische Informatik	Datenbanken	Schwerpunktveranstaltung
Diskrete Mathematik	Grundlagen der praktischen Informatik	Algorithmen und Datenstrukturen	Softwaretechnik	Kerninformatik – Wahlmodul	Wahlmodul
Grundlagen der Informatik und Programmierung	Schwerpunktveranstaltung	Telematik / Computernetzwerke	Kerninformatik – Wahlpflicht	Schwerpunktveranstaltung	Bachelorarbeit
Programmierkurs	Allgemeines Programmierpraktikum	Schwerpunktveranstaltung	Schwerpunktveranstaltung	Wahlmodul	
			Schwerpunktveranstaltung	Schlüsselkompetenz – Wahlmodul	
			Fachpraktikum		

Mathematische Grundlagen 36C	
Informatik Grundlagen 20C	Kerninformatik 40C
Wahlbereich 10C	Schlüsselkompetenzen 20C
Studienschwerpunkt 42C	Bachelorarbeit 12C

³<https://www.uni-goettingen.de/de/640717.html>

Zusammenfassung

- Hochleistungsrechnen ist Formel-1 für Computersysteme
- Wissenschaftliche Erkenntnis benötigt Hochleistungsrechnen
 - ▶ Mitarbeit an gesellschaftlich relevanten Themen
 - ▶ Erfüllende interdisziplinäre Zusammenarbeit
- Informatik formt die Basis-Technologie für das Informationszeitalter
- Informatik treibt unsere Gesellschaft an
- Informatik ist faszinierend und vielfältig
- Informatik ermöglicht uns persönliche Entfaltung
- Informatik bietet eine Zukunftsperspektive
- Jede:r kann Informatik lernen
- Wir freuen uns darauf, euch bei eurem Weg zu begleiten

Wir freuen uns über Ihr Feedback zu den Infotagen!

Jetzt an der kurzen Online-Umfrage teilnehmen und ein kleines Dankeschön am Infopoint erhalten.



Umfrage-Link auch auf der Webseite: <https://www.uni-goettingen.de/infotage>