Einführung in die Theoretische Informatik

Woche 1

Harald Zankl

Institut für Informatik @ UIBK
Wintersemester 2014/2015

Einleitung

Theoretische Informatik

Die Theoretische Informatik beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.

Ihre Inhalte sind Automatentheorie, Theorie der formalen Sprachen, Berechenbarkeits- und Komplexitätstheorie, aber auch Logik und formale Semantik sowie die Informations-, Algorithmen- und Datenbanktheorie.

http://de.wikipedia.org/ 2013

1. Automatentheorie
2. Theorie der formalen Sprachen
3. Berechenbarkeits- und Komplexitätstheorie
4. Logik und formale Semantik
5. Informations-, Algorithmen- und Datenbanktheorie

Handbook of Theoretical Computer Science

 +  = 2293 Seiten, 4 Kilogramm

Inhaltsverzeichnis Band „Algorithms and Complexity“

Machine models and simulations, A catalog of complexity classes, Machine-independent complexity theory, Kolmogorov complexity and its applications, Algorithms for finding patterns in strings, Data structures, Computational geometry, Algorithmic motion planning in robotics, Average-case analysis of algorithms and data structures, Graph algorithms, Cryptography, Algebraic complexity theory, Algorithms in number theory, The complexity of finite functions, Communication networks, VLSI theory, Parallel algorithms for shared-memory machines, General purpose parallel architectures

Inhaltsverzeichnis Band „Formal Models and Semantics“

Finite automata, Context-free languages, Formal languages and power series, Automata on infinite objects, Graph rewriting: an algebraic and logic approach, Rewrite systems, Functional programming and lambda calculus, Type systems for programming languages, Recursive applicative program schemes, Logic programming, Denotational semantics, Semantic domains, Algebraic specification, Logics of programs, Methods and logics for proving programs, Temporal and modal logic, Elements of relational database theory, Distributed computing: models and methods, Operational and algebraic semantics of concurrent processes

Inhalte der Lehrveranstaltung

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Formales Beweisen, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

Boolsche Algebra, Universelle Algebra, Logische Schaltkreise

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, Kontextfreie Sprachen

Einführung in die Berechenbarkeitstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen

Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare

Einführung in die Logik

Grundlage für: Logik (3. Semester); nützlich für Funktionale Programmierung (3. Semester) Logische Programmierung (Wahlmodul) und Automatisches Beweisen (Master)

Einführung in die Algebra

Grundlage für: Entwurf von Softwaresystemen (3. Semester); nützlich für Einführung in die technische Informatik (1. Semester)

Einführung in die Theorie der Formalen Sprachen

Grundlage für: Diskrete Mathematik (2. Semester); nützlich für Formale Sprachen und Automatentheorie, Compilerbau (Master)

Einführung in die Berechenbarkeitstheorie

wir müssen unsere Grenzen kennen

Einführung in die Programmverifikation

nützlich für Entwurf von Softwaresystemen (3. Semester)

Geschichte der Theoretischen Informatik

	Rechenmodelle	Digitalrechner	
	Turing Maschinen, Berechenbarkeitstheorie; Alan Turing	—	1930er
	Formale Sprachen, Automatentheorie	`Zuse Z3, ENIAC`	1940er
	Grammatiken, Grundlagen des Compilerbaus; Noam Chomsky	`UNIVAC`, Transistoren statt Röhren	1950er
	P vs. NP, Komplexitätstheorie; Stephen Cook	Minicomputer, integrierte Schaltkreise	1960er

Entschlüsselung der ENIGMA

- „Enigma“ ist griechisch für Rätsel
- deutsche Kodiermaschine eingesetzt im 2. Weltkrieg
- galt als unentzifferbar, Entschlüsselung benötigte etwa 8 Jahre
- Hauptakteure der Entschlüsselung: Rejewski & Turing
- manuelle Entschlüsselung erwies sich als nicht praktikabel (eigentlich unmöglich)
- Code wurde maschinell entschlüsselt
- wesentliche Werkzeuge: mathematische Analyse und Automatisierung
- *“It was thanks to Ultra that we won the war”* (W. Churchill)

Einführung in die Logik

Frage

Wie argumentieren wir im täglichen Leben?

Beispiel

Sokrates ist ein Mensch.	}	Prämisse ①
Alle Menschen sind sterblich.	}	Prämisse ②
Somit ist Sokrates sterblich.	}	Konklusion

Definition

- Schlussfiguren dieser Art heißen Syllogismen
- Syllogismen wurden bereits im antiken Griechenland untersucht

Fakt

Nicht die Wahrheit der Prämissen, oder der Konklusion, sondern die Wahrheit der Schlussfigur ist entscheidend.

Beispiel

Der Mond besteht aus grünem Käse.	}	Prämisse ①
Die Sonne geht im Westen auf.	}	Prämisse ②
Tirol liegt im Flachland.	}	Konklusion

Fakt

Alle Aussagen in dem Beispiel sind falsch; trotzdem ist die Schlussfigur wahr, da aus Falschem Beliebiges folgt.

Beispiel

Tirol ist bergig.	wahre Aussage
Die Sonne geht im Osten auf.	wahre Aussage
Also, besteht der Mond aus grünem Käse.	falsche Aussage

Fakt

Die Schlussfigur ist falsch, da aus Wahrem etwas Falsches gefolgert wird.

Modus Ponens

Beispiel

Wenn das Kind schreit, dann hat es Hunger.
Das Kind schreit.
Also, hat das Kind Hunger.

Fakt

Die Korrektheit dieser Schlussfigur ist unabhängig von den konkreten Aussagen.

Definition (*Modus Ponens*)

Wenn A, dann B.
A gilt.
Also, gilt B.