

Automatisches Beweisen—Vertiefung

Einführung

Christoph Zengler

Arbeitsbereich Symbolisches Rechnen
Prof. Dr. Wolfgang Kuchlin
Universität Tübingen

11. Oktober 2011

i Die Vorlesung

- **Zeit:** Dienstags, 14 c.t. – 16
- **Ort:** Hörsaal A104

-
- Anrechenbar für Master / Bachelor: *Wahlpflichtbereich Theoretische Informatik*
 - 4 LP (Diplom 3 SWS)

i Die Übung

- **Zeit:** Dienstags, 13 c.t. – 14
- **Ort:** Hörsaal A104
- **Tutor:** Christoph Zengler
- **Start:** Dienstag, 18. Oktober 2011

i Die Klausur

- **Termin:** Dienstag 31. Januar 2012, 14.00 - 16.00
- **Art:** Mündlich/Schriftlich je nach Teilnehmerzahl

-
- BSc / MSc Klausur wird *nicht* identisch sein

i Benotung

- Klausur 70%
- Übung 30%

i Das Praktikum

- Eigenes Modul, optional zur Vorlesung (kann auch später belegt werden)
 - 4 LP (Diplom 2 SWS)
 - Anrechenbar für Wahlpflichtbereich Informatik
-
- Implementierung der Verfahren aus der Vorlesung in Java
 - Java Model Checker mit graphischer Oberfläche
-
- 2 Wochen als Block am Ende des Semesters
 - Termin kann flexibel auf Teilnehmer angepasst werden

Der Arbeitsbereich Symbolisches Rechnen



Susanna Uresch
(Sekretariat)



Prof. Dr. Wolfgang Küchlin

Arbeitsbereich Symbolisches Rechnen



Dipl. Inform.
Christoph Zengler



Dipl. Inform.
Andreas Kübler

Steinbeis Transfer Zentrum



Dipl. Inform.
Martin Rathgeber



PD Dr.
Reinhard Bündgen

PD Dr.
Wolfgang Blochinger



Sprechstunde Prof. Küchlin
Donnerstags 10.00 - 11.30

Sprechstunden Sekretariat
Montag 14 - 16
Dienstag 10 - 12
Mittwoch 10 - 12
Donnerstag 14 - 16

Das Vorlesungsangebot

rot: wird dieses Semester gelesen

	Bachelor	Master
Logik	Automatisches Beweisen Grundlagenpraktikum AB	Automatisches Beweisen—Vertiefung Vertiefungspraktikum AB SAT Solving & Anwendungen Termersetzungssysteme
Systeme	Betriebssysteme Praktikum zu Betriebssysteme	Linux Parallele Systeme

- Dazu meistens Seminare & Prosmeminare aus den jeweiligen Gebieten, z.B.
 - Automatische Entscheidungsverfahren in der Softwareverifikation (Seminar, WS 2009/10)
 - Knowledge Representation & Reasoning (Proseminar, WS 2010/11)
 - Konfiguration (Seminar, SS 2011)

- BSc & Studienarbeiten:

<http://www-sr.informatik.uni-tuebingen.de/bsc>

- MSc & Diplomarbeiten:

<http://www-sr.informatik.uni-tuebingen.de/msc>

Letzte Arbeiten

- Non-CNF Model Counting und Anwendungen in der Fahrzeugkonfigurationsprüfung (DA, 2010)
- Verifikation eines Linux-Gerätetreibers (DA, 2010)
- Entwurf und Implementierung eines Codeanalyseframeworks für Eclipse (DA, 2011)
- Formale Verifikationsmethoden für C++ Programme (DA, 2011)
- Implementierung eines NNF SAT Solvers (SA, 2011)
- Prüfung intervallbasierter automobiler Konfigurationsdaten (MSc, 2011)

Ein paar Worte zu Vorlesung & Übung

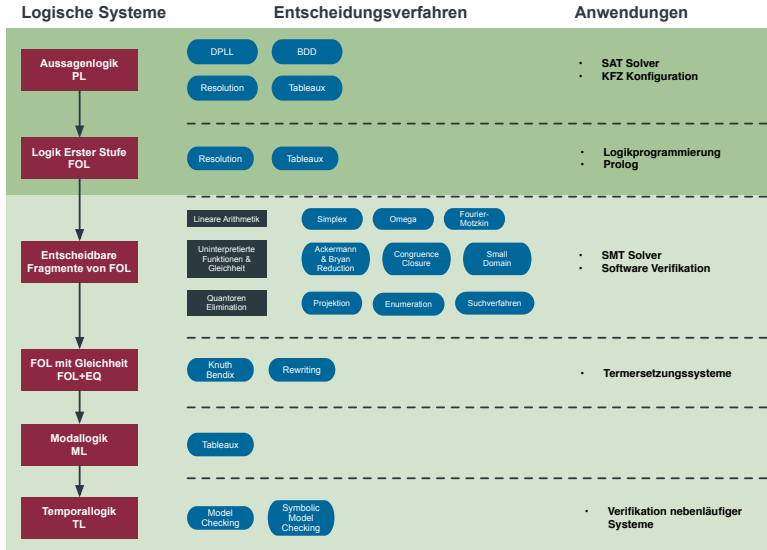
Vorlesung

- Folien zur Vorlesung am Abend vorher im Netz unter <http://www-sr.informatik.uni-tuebingen.de/abv>
- Folien sind Stoffsammlung und alleine *nicht* ausreichend
- Viele Erklärungen, Beispiele, Demos *nur* in der Vorlesung (Tafel & Kreide!)

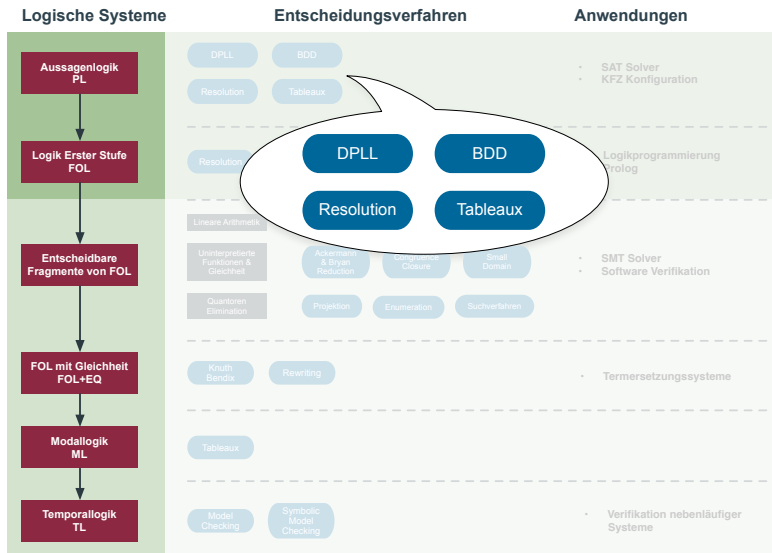
Übung

- Übungsblätter immer Mittwochs Nachmittag im Netz
- Abgabe Montags – 14.00 vor Raum C 103 oder an christoph@zengler.eu
- Teams bis zu 2 Personen möglich
- Übungen sind unerlässlich zum Bestehen der Klausur

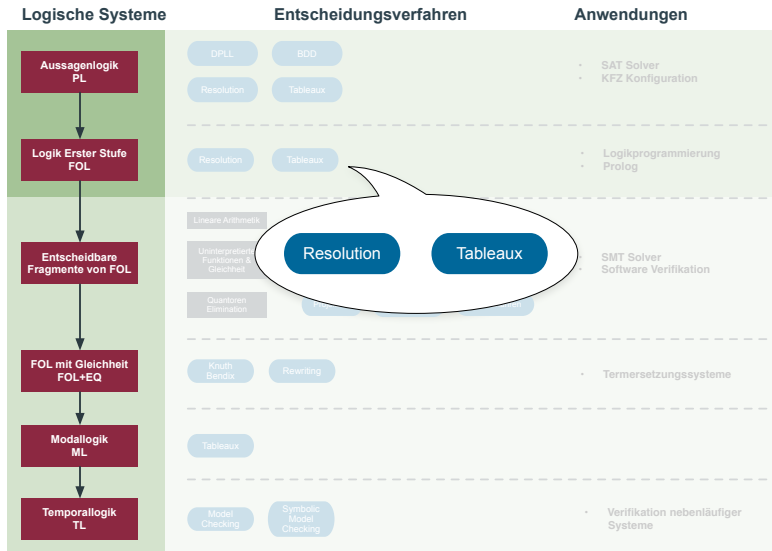
Stoffübersicht—The Big Picture



Stoffübersicht—The Big Picture



Stoffübersicht—The Big Picture



Einführung

Christoph
Zengler

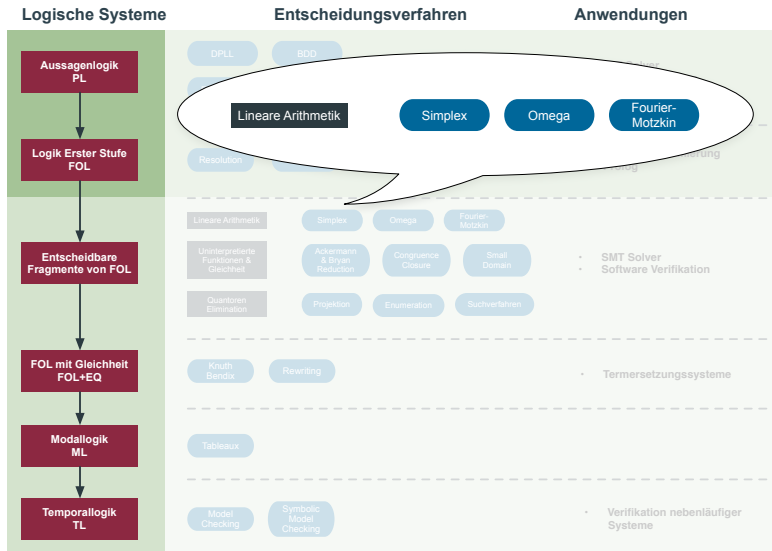
Einführung

Organisatorisches
Der Arbeitsbereich

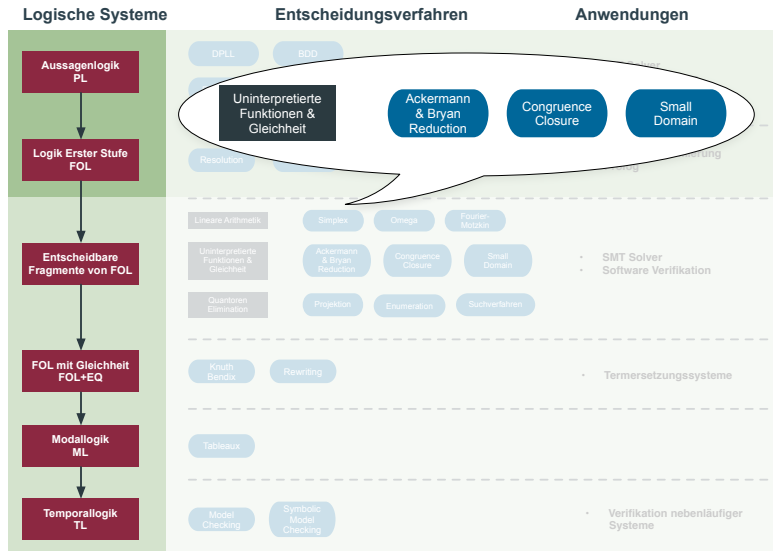
Vorlesungs Ablauf

Warthog

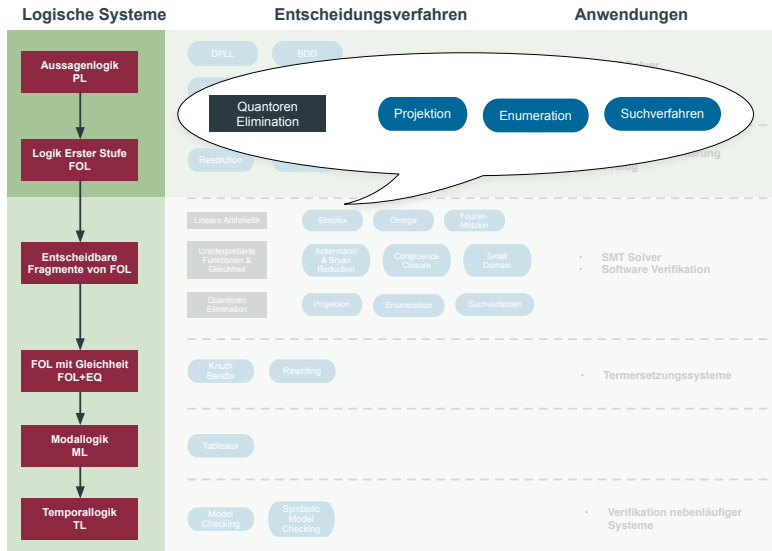
Stoffübersicht—The Big Picture



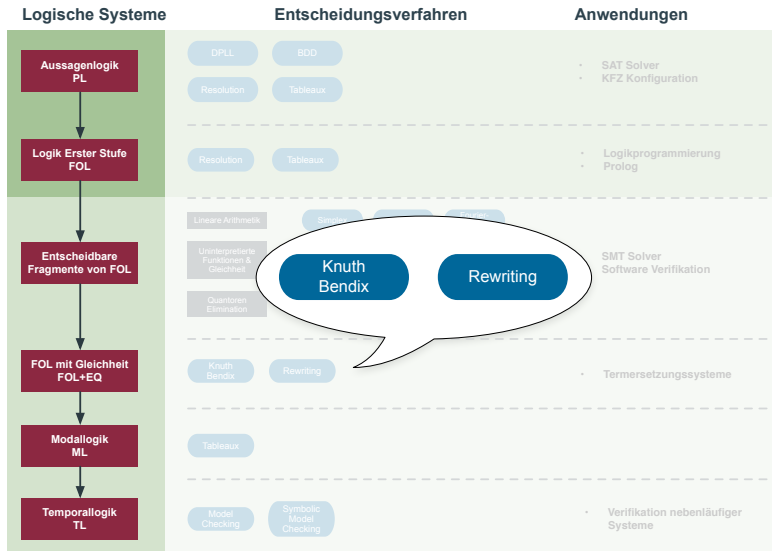
Stoffübersicht—The Big Picture



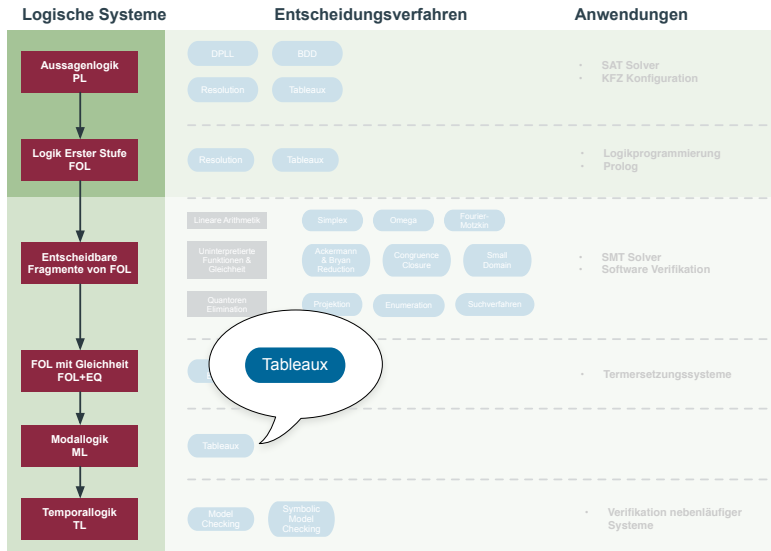
Stoffübersicht—The Big Picture



Stoffübersicht—The Big Picture



Stoffübersicht—The Big Picture



Einführung

Christoph
Zengler

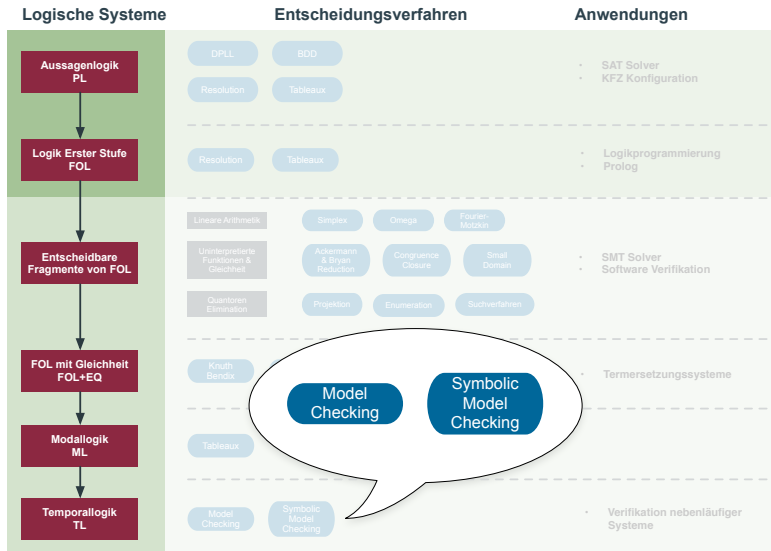
Einführung

Organisatorisches
Der Arbeitsbereich

Vorlesungs Ablauf

Warthog

Stoffübersicht—The Big Picture



Einführung

Christoph
Zengler

Einführung

Organisatorisches
Der Arbeitsbereich

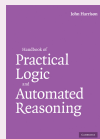
Vorlesungs Ablauf

Warthog

- ① Verschiedene Logische Systeme mit
 - Syntax & Semantik
 - Normalformen
 - Entscheidungsverfahren
 - Theoretische Ergebnisse
- ② Anwendung auf “echte” Probleme, wie z.B.
 - Software Verifikation
 - Software Konfiguration
 - Verifikation nebenläufiger Systeme
- ③ Codierung von Problemen in logische Formeln
- ④ Ein Gefühl dafür, was ...
 - ... mit bestimmten logischen Systemen geht, und was nicht
 - ... welches System in welcher Situation zu bevorzugen ist
- ⑤ Automatisches Beweisen ist cool 😊



Literaturhinweis



John Harrison

Handbook of Practical Logic and Automated Reasoning

Cambridge University Press, 1. Auflage 2009

ISBN: 978-0521899574



Michael Huth & Mark Ryan

Logic in Computer Science

Cambridge University Press, 2. Auflage 2004

ISBN: 978-0521543101



Martin Kreuzer & Stefan Kühling

Logik für Informatiker

Pearson Studium, 1. Auflage 2006

ISBN: 978-3827372154

Einführung

*Christoph
Zengler*

Einführung

Organisatorisches
Der Arbeitsbereich

Vorlesungs Ablauf

Warthog

Abschließende Motivation—Warum Logik?

Man bekommt ganz gute Arbeitgeber

- Intel (z.B. John Harrison)
- Google (z.B. Knot Pipatsrisawat)
- IBM (z.B. Reinhard Bündgen)
- Microsoft (z.B. Leonardo de Moura, Nikolaj Bjørner)

Man kann unerwartete Dinge damit machen

- Der Spielplan der holländischen Fußball-Ligen wird mit einem SMT Solver berechnet
- Updateabhängigkeiten in Eclipse werden mit SAT Solver berechnet

Man kann damit wirklich Geld verdienen

- **Mathematical Logic Finds Unexpected Application on Wall Street:** ... *a few logicians were offered seven figure salaries* ...
- Lösung des SAT Problems: \$1.000.000

Warthog

- Ein **Logik Framework** für verschiedene logische Systeme (z.B. Aussagenlogik, Logik erster Stufe, Temporallogik, Higher Order Logik,...)
- Initiiert im Sommer 2011 von Andreas Kübler und Christoph Zengler
- **Einsatzbereich:** Lehre & Industrielle Anwendungen
- **Programmiersprachen:** Scala, Java, native Teile in C und C++
- **Ziel:** Vereinigung vieler Systeme, Algorithmen und Formate in einem einheitlichen Framework

Was ist bisher implementiert?

① Aussagenlogik

- Evaluation & Wahrheitstabellen
- Normalformen: NNF, CNF, DNF
- Einfache Simplifikationen
- BDDs (eigene Implementierung)
- SAT Solver (PicoSAT & MiniSAT Anbindung)
- Cardinality Constraints, Langford Pairings, Pigeon Hole

② Logik erster Stufe

- Evaluation mit endlichen Domänen
- Normalformen: PNF
- Skolemisierung

③ Interaktive Benutzung über Scala REPL

④ Deine nächste Abschlussarbeit?

-
- Demos in der Vorlesung
 - Source Code ab ca. Anfang 2012 öffentlich zugänglich