Karlsruher Institut für Technologie

Deductive Software Verification – The KeY Book
Haupt- und Proseminar in SoSe 2017

Bernhard Beckert | 02.05.17

INSTITUT FÜR THEORETISCHE INFORMATIK, KIT

KeY-Projekt

www.key-project.org

Konsortium

- Bernhard Beckert
  Karlsruhe Institute of Technology
- Reiner Hähnle
  TU Darmstadt
- Wolfgang Ahrendt
  Chalmers Univ., Göteborg

KeY-Projekt

www.key-project.org

Deduktive Verifikation

- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Projekt

www.key-project.org

Deduktive Verifikation

- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Tool

- Deduktive Regeln für alle Java-Features
- Symbolische Programmausführung
- Halb-automatisch
  (beides wichtig: Automatisierung und Usability)

KeY-Projekt

www.key-project.org

Deduktive Verifikation

- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Tool

- Deduktive Regeln für alle Java-Features
- Symbolische Programmausführung
- Halb-automatisch
  (beides wichtig: Automatisierung und Usability)

KeY-Projekt

www.key-project.org

Deduktive Verifikation

- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Tool

- Deduktive Regeln für alle Java-Features
- Symbolische Programmausführung
- Halb-automatisch
  (beides wichtig: Automatisierung und Usability)

KeY-Projekt

www.key-project.org

Deduktive Verifikation

- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Tool

- Deduktive Regeln für alle Java-Features
- Symbolische Programmausführung
- Halb-automatisch
  (beides wichtig: Automatisierung und Usability)

KeY-Projekt

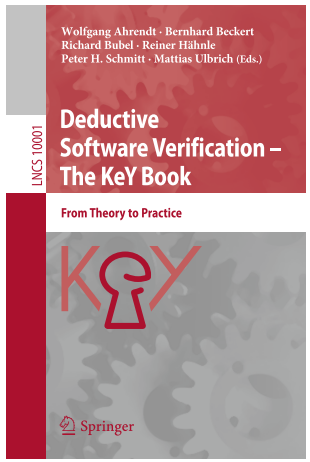
www.key-project.org

Deduktive Verifikation

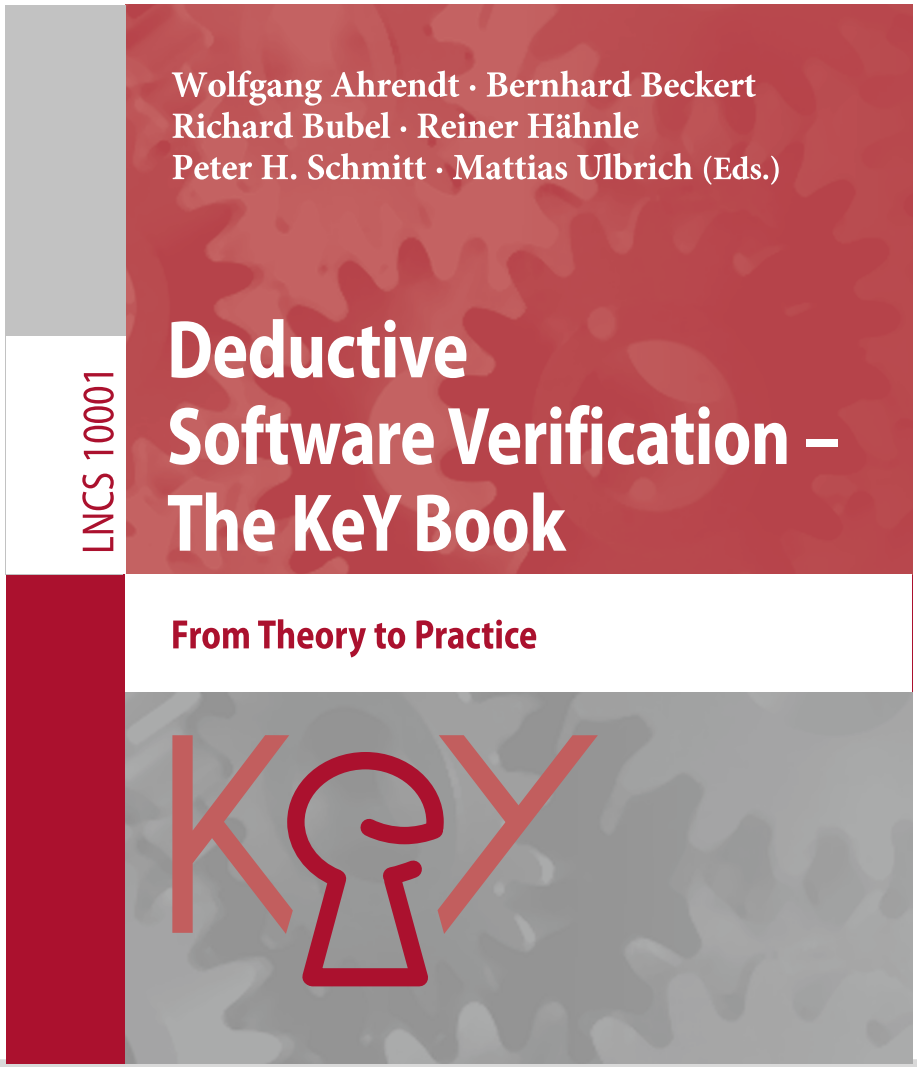
- Java
- Spezifikation:
  Java Modeling Language
- Source Code Level

KeY-Tool

- Deduktive Regeln für alle Java-Features
- Symbolische Programmausführung
- Halb-automatisch
  (beides wichtig: Automatisierung und Usability)



Deductive Software Verification



Organisatorisches I

Aufgabenpunkte: Pro- und Hauptseminar

- Verstehen des Stoffes
- Auswahl der Themen, die präsentiert werden sollen
- Planung des Vortrags
- Kurzpräsentation der Gliederung (5 Minuten)
- Erstellen der Folien
- Vortrag (30 Minuten)
- Schriftliche Ausarbeitung (~10 Seiten)

Hauptseminar

- Erarbeitung und Darstellung des Themas über Buchkapitel hinaus
- $\Rightarrow$ ergänzende Literaturrecherche

Organisatorisches II

Anmeldung

- Entscheidung Teilnahme / Themenzuordnung bis Ende der Woche, 05.05.17, 14 Uhr (Reihenfolge in Anmeldungsreihenfolge)
- E-Mail an `meinhart@kit.edu` mit Präferenzliste für die Themen
- bis 13.06.17: Prüfungsanmeldung
  Beachten Sie bitte das **Handout**

Organisatorisches III

Zwischenpräsentation

- je 5 Minuten
- ca. 2. Juni-Woche
- **Abgabe der Folien:** 1 Woche voher

Hauptpräsentation

- Blockseminar, ca. 2 Tage am Semesterende
- 30 min. + 10 min. Fragen
- **Abgabe der Folien:** 1 Woche vor dem 1. Termin

Ausarbeitung

- **Abgabe:** 31. Sept. 2017 (Springer LNCS, ca. 10 Seiten)

Themen – Proseminar

Kapitel 5	Theories
Kapitel 7	Formal Specification with the Java Modeling Language
Kapitel 11	Debugging and Visualization
Kapitel 12	Proof-based Test Case Generation
Kapitel 17	KeY-Hoare
Kapitel 19	Verification of Counting Sort and Radix Sort

Liste der Themen mit Betreuer auf
`https://formal.iti.kit.edu/teaching/Proseminar-SS17`

Themen – Hauptseminar

Kapitel 2	First-Order Logic for Java
Kapitel 3	Dynamic Logic for Java
Kapitel 6	Abstract Interpretation
Kapitel 8	From Specification to Proof Obligations
Kapitel 9	Modular Specification and Verification
Kapitel 13	Information Flow Analysis

Liste der Themen mit Betreuer auf
`https://formal.iti.kit.edu/teaching/Seminar-SS17`