

# Verifikation nebenläufiger Programme

## Vorlesung in Sommersemester 2000

Lehrstuhl für Softwaretechnologie, Prof. Dr. Willem-Paul de Roever

**Zeit:** Mo. 9–11, Mi. 9–11 ct.

**Fachgebiet:** theoretische und praktische Informatik

**Zielgruppe:** Studierende im Hauptstudium

Es ist mittlerweile klar, daß die meisten Spezifikations- und Verifikationsmethoden die Grenzen ihrer Anwendbarkeit erreicht haben. Algorithmische Methoden können endliche Systeme nur bis zu einer bestimmten Größe überprüfen, während deduktive Methoden nur für Systeme geringer Komplexität geeignet sind, weil sie von intensiver Benutzerinteraktion abhängig sind. Die einzige Hoffnung, formale Verifikation für Probleme von industrieller Größe einzusetzen, baut auf die Anwendung zweier wichtiger Prinzipien: *Kompositionalität* und *Abstraktion*.

Kompositionelle Verifikationsmethoden ermöglichen es, Eigenschaften eines zusammengesetzten Systems aus den Spezifikationen seiner Teile herzuleiten, ohne interne Details der Komponenten zu verwenden. Die zwei Hauptfragen für kompositionelle Verifikationsmethoden sind also:

1. Wie zerlegt man eine globale Spezifikation in Teilspezifikationen der Komponenten?
2. Wie schließt man aus der Erfüllung der Teilspezifikationen auf die Erfüllung der globalen Spezifikation?

Nach der Behandlung der Grundlagen traditioneller, nicht-kompositioneller Beweismethoden liegt der Schwerpunkt der Vorlesung auf der Entwicklung kompositioneller Beweismethoden für nebenläufige Programme.

Da das begleitende Manuskript dieses Jahr als Buch veröffentlicht werden wird, soll es in dieser Vorlesung ausgiebig auf Konsistenz geprüft werden. *Studenten, die sich hieran beteiligen möchten, werden gebeten, an der Vorlesung aktiv teilzunehmen: der Beitrag wird im Vorwort des Buches erwähnt.*

Bei Fragen zur Vorlesung: `{wpr|kba}@informatik.uni-kiel.de`. Weitere Informationen, auch wie man das Buchmanuskript erhalten kann, finden sich unter:

<http://www.informatik.uni-kiel.de/inf/deRoever/SS00/Verifikation>

## Literatur