Übung zur Vorlesung

Automatisierte Logik und Programmierung

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Sommersemester 2004

Blatt 6 — Abgabetermin: —

Dieses letzte Übungsblatt dient im wesentlichen der Selbstkontrolle bei der Rekapitualtion der angesprochenen Themen.

Aufgabe 6.1 (Rückblick, 1. Teil)

Die folgenden Aufgaben sind vorgesehen als Kontrollfragen zur Überprüfung des eigenen Kenntnisstandes. Sie entsprechen in ihrer Thematik dem Spektrum einer mündlichen Prüfung. Die Antworten sind größtenteils im Skript bzw. auf den Folien enthalten, allerdings nur selten an auffälliger Stelle. Versuchen sie, diese zunächst ohne Ihre Unterlagen zu beantworten.

6.1–a Welche 5 prinzipiellen Möglichkeiten einer automatischen Unterstützung für formales Beweisen gibt es?

6.1–b Welche Doppelrolle spielt die Sprache ML für interaktiver Beweissysteme?

6.1–c Schreiben Sie ein ML-Programm `prim`, welches einen Primzahltest durchführt.

6.1–d Durch welche Datenstruktur werden Sequenzenbeweise in ML repräsentiert? Nennen sie auch die wichtigsten Zugriffs- und Manipulationsoperatoren.

6.1–e Welche Besonderheit besitzt der Termeditor von NuPRL und welche Vorteile ergeben sich daraus?

6.1–f Welcher Mechanismus ist erforderlich, um konservative Erweiterungen der Typentheorie durch Abstraktionen fehlerfrei zu ermöglichen?

6.1–g Erklären Sie die besondere Rolle von Taktiken bei der Beweisführung.

6.1–h Wodurch unterscheiden sich Verfeinerungs- und Transformationstaktiken?

6.1–i Warum sind Beweise, die mit Taktiken geführt werden, immer korrekt?

6.1–j Welche Voraussetzungen müssen erfüllt sein, um eine Entscheidungsprozedur in einem Beweissystem einzusetzen?

6.1–k Warum gibt es keine Entscheidungsprozedur für die gesamte Arithmetik?

6.1–l Welche Art von Problemen entscheidet die Prozedur `arith`?

6.1–m Erklären Sie die grundsätzliche Arbeitsweise der Prozedur `arith`.

6.1–n Erklären Sie die grundsätzliche Arbeitsweise der Prozedur `equality`.

6.1–o Welche Probleme besitzen Beweissysteme, die sich ausschließlich auf Entscheidungsprozeduren und Benutzerinteraktion stützen?

6.1–p Welche Arten von Beweisführung gibt es für die Logik erster Stufe und wie funktionieren sie?

6.1–q Beschreiben Sie die Funktionsweise von JProver.

Aufgabe 6.2 (Rückblick, 2.Teil)

6.2–a Warum kann es ein vollständiges Programmsynthesesystem niemals geben?

6.2–b Beschreiben Sie die Phasen einer formalen Entwicklung von Software.

6.2–c Welche Komponenten von Programmen und Spezifikationen sind für Programmsynthese von Bedeutung?

6.2–d Geben Sie eine präzise Definition der Begriffe "Programmkorrektheit" und "Erfüllbarkeit von Spezifikationen" auf der Basis der oben genannten Komponenten.

6.2–e Welche grundsätzlichen Paradigmen für Programmsynthese kennen Sie? Beschreiben Sie die grundsätzlichen Unterschiede bei der Darstellung des Problems, der Art der logischen Inferenzen bei der Problemlösung und der Konstruktionsmethode für den endgültigen Algorithmus.

6.2–f Welches Problem führt dazu, daß das Prinzip "Beweise-als-Programme" in der Praxis bisher nur von geringer Bedeutung ist?

6.2–g Welcher Programmierstil wird durch eine Synthese mit Formeltransformationen besonders unterstützt?

6.2–h Was ist die charakterisierende Eigenschaft einer Algorithmentheorie?

6.2–i Beschreiben Sie das generelle Verfahren zur Synthese von Algorithmen mit Hilfe von Algorithmenschemata. Wodurch wird dieses Vorgehen gerechtfertigt?

6.2–j Welche Vorteile hat die Synthese mit Algorithmenschemata gegenüber anderen Verfahren?

6.2–k Beschreiben sie die allgemeine Struktur von Globalsuchalgorithmen und die Voraussetzungen für ihre Korrektheit.

6.2–l Beschreiben Sie die allgemeine Strategie zur Erzeugung von Globalsuchalgorithmen

6.2–m Auf welche Art könnte man Synthesestrategien in das formale Konzept typentheoretischer Schlußfolgerungen integrieren?

6.2–n Beschreiben Sie die allgemeine Struktur von Divide & Conquer Algorithmen und die Voraussetzungen für ihre Korrektheit.

6.2–o Beschreiben Sie eine Strategie zur Erzeugung von Divide & Conquer Algorithmen

6.2–p Beschreiben Sie die allgemeine Struktur von Lokalsuchalgorithmen und die Voraussetzungen für ihre Korrektheit.

6.2–q Welche Techniken zur Optimierung von synthetisierten Algorithmen kennen Sie?