Theorie und Implementation objektorientierter Programmiersprachen
Theoretische Informatik, LMU München
Andreas Abel

Prüfungsstoff

Für die mündliche Prüfung soll folgender Stoff beherrscht werden.

1. Begriffe
   - Term
   - Induktive Definition
   - Inferenzregel
   - Herleitung
   - Definition einer Funktion durch Rekursion
   - Induktiver Beweis
   - Reduktion
   - Auswertung
   - Small-step-Semantik
   - Big-step-Semantik
   - Berechnungsregel
   - Kongruenzregel
   - Typ
   - Typisierung
   - Typprüfung
   - Typinferenz
   - Typsicherheit
   - Typerhaltung unter Reduktion/Auswertung
   - Fortschritt
   - Einführungsregel
   - Beseitungsregel
   - entscheidbar und unentscheidbar
   - Kontext (im getypten Lambda-Kalkül)
2. Sprachen (jeweils Syntax, cbv-Auswertung, Typisierung)
   - Ungetypter Lambda-Kalkül
     - de-Bruijn-Repräsentation von Termen
   - Getypter Lambda-Kalkül
   - Erweiterungen des getypten Lambda-Kalküls
     (a) Produkte
     (b) Einelementiger Typ
     (c) Sequenzen und `let`
     (d) Datensätze (Records)
     (e) Rekursion mit `fix`
   - Referenzen
   - Untertypen
3. Sonstiges
   - Programmieren im ungetypten Lambda-Kalkül
   - Objektorientiertes Programmieren in funktionaler Sprache mit Referenzen
   - Beweisen durch strukturelle Induktion

Für die Prüfung **unwichtig** sind:

- Historische Details.
- Syntax von Haskell, SML oder Java.
- Die *Namen* der vorgestellten Inferenzregeln.
- Parsen.
- Verschiedene Auswertungsstrategien für den Lambda-Kalkül.
- Substitution, Shifting oder Auswertung in de-Bruijn-Repräsentation.
- Übersetzung von einer Sprache in eine andere ($\ulcorner t \urcorner$, wurde einmal in der Übung erwähnt).

Stand: 08. Juli 2002