

Fortgeschrittene Programmierung / SS'13

Michael Hanus

20. Juni 2013

Detaillierter Vorlesungsverlauf

- 9.4.: Einführung in die funktionale Programmierung:** Variablenbegriff, Programm, Programmablauf, Funktionsdefinitionen, Ausdrücke, Beispiele `square/min/fac`, Auswertungsmöglichkeiten, Fibonacci-Zahlen, lokale Definitionen, Layout-Regel
- 11.4.:** Vorteile lokaler Definitionen, Basisdatentypen, Typannotationen, algebraische Datentypen (Aufzählungstypen, Verbundtypen, gemischte Typen, Listen), Operatoren, Vergleich und Ausgabe von Daten (`deriving (Eq,Show)`),
- 16.4.** Polymorphe Funktionen (`length`, `++`), `last`), Definition polymorpher algebraischer Datentypen, `Maybe`, Binärbäume, `String`, `Either`, Tupel (`fst`, `snd`, `zip`),
- 18.4.** `unzip`, Pattern Matching (Patternaufbau, `case`-Ausdrücke), Guards, Funktionen höherer Ordnung, anonyme Funktionen
- 23.4.** partielle Applikation, Currying, Sections, generische Programmierung (`map`, `foldr`, `filter`
- 25.4.** `foldl`), Kontrollstrukturen als Funktionen höherer Ordnung (`while`), Funktionen als Datenstrukturen (Implementierung von Feldern), wichtige Funktionen höherer Ordnung (Komposition, `flip`, `curry/uncurry`, `const`), Motivation und Struktur von Typklassen
- 30.4.** vordefinierte Funktionen in Typklassen, Standardklassen, `deriving`, Klasse `Read` und Funktionen `read` und `reads`, Lazy Evaluation (`from`, `primes`, `fibs`)
- 2.5.** Graphreduktion, Idee der Ein-/Ausgabe I/O-Aktionen, `do`-Notation
- 14.5.** Beispiel Ausgabe von Zwischenergebnissen, list comprehensions, Module
- 16.5. Einführung in die Logikprogrammierung:** Motivation, Verwandtschaftsbeispiel, Prolog-Programme, Fakten, Regeln, Anfragen, Prolog-Syntax (Zahlen, Atome)
- 21.5.** Prolog-Syntax (Strukturen, Listen), Operatoren, Variablen, Rechnen mit Listenstrukturen, Gleichheit von Termen; Programmieretechnik Aufzählung des Suchraumes (Färben einer Landkarte, Sortieren von Zahlenlisten)
- 23.5.** Programmieretechnik Musterorientierte Wissensrepräsentation (`append`), Verwendung von Relationen, Arithmetik in Prolog (`is`, Fakultätsfunktion), arithmetische Constraints, Beispiel Schaltkreisanalyse

- 28.5.** Beispiel Hypotheksberechnung, Constraint-Programmierung über endlichen Bereichen, allgemeines Vorgehen, send-more-money-Beispiel, 8-Damen-Problem, verbesserte Labeling-Strategien
- 30.5.** Weitere FD-Constraints, CP in anderen Sprachen; Rechnen in der Logikprogrammierung: einfaches Resolutionsprinzip, Substitution, Unifikation, mgu, Unifikationsalgorithmus, occur check, Komplexität
- 6.6.** allgemeines Resolutionsprinzip (SLD-Resolution), Beweisstrategie von Prolog, Endlosschleifen, Cut-Operator, Fallunterscheidung, Negation als Fehlschlag
- 11.6.** Probleme der Negation
Java Generics: parametrisierte Containerklasse, Typparameter, eingeschränkte Typparameter, Wildcards, beschränkte Wildcards, Hinweise zur Implementierung von Generics
- 13.6.** Nebenläufige Programmierung in Java: grundlegende Begriffe, Synchronisationsproblem, Semaphore, Dining Philosophers, Klasse `Thread`, Interface `Runnable`, Eigenschaften von Thread-Objekten
- 18.6.** Synchronisation von Threads in Java, `synchronized`-Methoden, `synchronized`-Anweisung, synchronisierte Methoden vs. synchronisierte Anweisungen, synchronisierte Collections, `wait`, `notify`, `notifyAll`, Kommunikation zwischen Threads
- 20.6.** sinnvolle Benutzung von `wait`, `notify`, `notifyAll`, einelementiger Puffer, Benutzung von Synchronisationsobjekten, Beenden und Unterbrechen von Threads, `InterruptedException`, Serialisierung von Daten
- 25.6.** Idee von RMI, Parameterübertragung, Serverseite, Clientseite, RMI-Registrierung