Deklarative Programmiersprachen
WS 2015/16

Michael Hanus

4. Februar 2016

Detaillierter Vorlesungsverlauf

27.10.: Einführung: Entwicklung von Sprachkonzepten, referenzielle Transparenz, Vorteile im Vergleich zu imperativen Sprachen (Optimierung, Parallelisierung, Zuverlässigkeit, Lesbarkeit), Beispiel Quicksort, Klassifikation von Programmiersprachen, Anwendungen deklarativer Programmiersprachen

29.10.: Beispiele: Adresssuche in HTML-Seiten, Su Doku-Löser mit Web-Interface; Funktionale Programmierung: Ausdrücke, Funktionsdefinitionen, Auswertung von Ausdrücken, Redex, strikte und nicht-strikte Sprachen, Fallunterscheidung, bedingte Regeln, Muster, Spezifikationssprache vs. deklarative Programmiersprache, Newtonsches Approximationsverfahren, Gültigkeitsbereich, lokale Deklarationen, Layout-Regel

3.11.: strenge Typisierung, Datentypen, Basistypen, strukturierte Typen (Listen, Zeichenketten, Tupel), funktionale Typen, Curryfizierung, benutzerdefinierte Datentypen, *Pattern Matching*: Vorteile, erlaubte Muster

5.11.: Auswertungsverhalten bei Mustern (Konjunktion, Paralleles Oder), `case`-Ausdrücke, Übersetzung muster-orientierter Funktionsdefinitionen in `case`-Ausdrücke

10.11.: Problem überlappender Regeln, Funktion `diag`, prinzipielles Problem sequentieller Auswertungsstrategien, Uniforme Funktionsdefinitionen, Reihenfolgeunabhängigkeit
Funktionen höherer Ordnung: Motivation, Programmierschema `map`, partielle Applikation (auch bei Infixoperatoren), `curry`, `uncurry`, `foldr`, `filter`, `rmdups`, Quicksort

12.11.: *Typsystem:* streng getypte Sprache, Typ, Typfehler, schwach getypte Sprache, ad-hoc Polymorphismus, parametrischer Polymorphismus, Typinferenz, typkorrekt, Typausdrücke, (Typ-)Substitution, Typinstanz, Typschema, generische Instanz, Typannahme, Inferenzsystem zur Typprüfung

17.11.: Typisierung von `twice`, allgemeinster Typ, Typprüfung für mehrere Funktionen, Typinferenz, Vorgehen, Berechnung allgemeinster Unifikatoren nach Martelli/Montanari

19.11.: Typinferenz für mehrere Funktionen, statischer Aufrufgraph, Grenzen der Typisierung: Selbstanwendung, `funsum`
Lazy Evaluation: Redex, Normalform, LO-Reduktion, schwache Kopfnormalform, unendliche Datenstrukturen, `from`, `fibs` Primzahlberechnung mittels Sieb des Eratosthenes

24.11.: *Reduktionssysteme:* Reduktionssystem, Reduktionsrelationen, reduzierbar, irreduzibel, Normalform, Church-Rosser, konfluent, lokal konfluent, terminierend, Noethersch, Newman-Lemma, Motivation Termersetzungssysteme (Gruppenaxiome), Signatur, Sorte, Funktionssymbol, Variable, Term, Grundterm, linearer Term, Termersetzungssystem

26.11.: Substitution, Position, Teilterm, Ersetzung, Vergleich von Termpositionen, Reduktionsschritt, Redex kritisches Paar, Kritisches-Paar-Lemma, (schwach) orthogonale Termersetzungssysteme, Reduktionsstrategie (sequenzielle, normalisierende)

1.12.: Reduktionsstrategien LI/LO/PO, linksnormale Termersetzungssysteme, Konstruktor, konstruktorbasiertes Termersetzungssystem, definierender Baum, induktiv-sequenziell, Reduktionsstrategie φ, vollständig definierte Funktionen, Eigenschaften von φ

3.12.: Motivation zur Logikprogrammierung: Verwandschaftbeispiel, Aspekte der logisch-funktionalen Programmierung Variablen in initialen Ausdrücken, Extravariablen in Regeln, `failed`, nichtdeterministische Operationen ("`?`")), Logiksprachen als Spezialfall logisch-funktionaler Sprachen, Spiel 24

8.12.: Reguläre Ausdrücke (Darstellung und Semantik), reguläre Ausdrücke (Semantik, Matching und `grep`)
Definition von Narrowing, Narrowing als Verallgemeinerung von Reduktion, Gleichungen, gültige Gleichungen, Lösungen von Gleichungen, Korrektheit und Vollständigkeit von Narrowing (Satz von Hullot)

10.12.: Vollständigkeit bzgl. normalisierter Substitutionen, reflexive Gleichheit und deren konstruktive Definition, Logikprogrammierung (Prolog) als Spezialfall der logisch-funktionalen Programmierung, Extravariablen, bedingte Regeln (Transformation in unbedingte), Resolution als Spezialfall von Narrowing, Backtracking, Färben einer Landkarte, Graph zeichnen (Haus vom Nikolaus)

15.12.: Strikte Narrowing-Strategien: Innermost Narrowing, Vollständigkeit, Innermost Basic Narrowing, Rückweisung (rejection), Normalisierendes Innermost Narrowing; Lazy Narrowing-Strategien: Outermost Narrowing

17.12.: Lazy Narrowing, Nachteile, Idee von Needed Narrowing, Steuerung von Needed Narrowing durch definierende Bäume, formale Definition von Needed Narrowing,

Eigenschaften (Vollständigkeit, Optimalität, Determinismus), erweiterte definierende Bäume für überlappende Regeln, Definition und Eigenschaften von Weakly Needed Narrowing

7.1.: Nichtdeterministische Operationen: Realisierung, intuitive Bedeutung, Anwendung (permutation sort), operationale Vorteile, semantische Probleme (call-time choice vs. run-time choice)
Residuation: operationale Idee, Vorteile, Anschluss externer Funktionen mittels Residuation, Nachteile von Residuation (Unvollständigkeit, unendliche Berechnungen mit verzögerten Constraints)

12.1.: Kombination von Residuation und Narrowing (flex/rigid branches), Strategie für diese Kombination, Auswertungsstrategie von Curry (optimale definierende Bäume), Gleichheitstest, Gleichheitsconstraint, bedingte Regeln, Funktionen höherer Ordnung (rigides `apply`), Modellierung nebenläufiger Objekte (Bankkonto)

14.1.: Erweiterungen: Constraint-Programmierung, CLP(R) (Hypothekenberechnung in Curry), CLP(FD): send-more-money-Beispiel, vordefinierte FD-Constraints, SuDoku-Löser
Eingekapselte Suche: Motivation, `findall`, Probleme, starke und schwache Einkapselung

19.1.: Probleme der starken und schwachen Einkapselung, rigide Einkapselung für logische Variablen, Set Functions, Datentyp `Values` und seine Operationen, Beispiel kürzeste Wegesuche

26.1.: Beispiel 8-Damen-Problem mit Set Functions, Funktionale Muster: Probleme der strikten Gleichheit bei `last`, Definition von `last` mit funktionalen Muster, deklarative Bedeutung funktionaler Muster, Ebenenabbildung und stratifizierte Programme, Unifikation "`=:<=`" für funktionale Muster, Beispiele `perm`, `descending`, `sort`

28.1.: Beispiele `lengthUpToRepeat`, Vereinfachung arithmetischer Ausdrücke
GUI-Programmierung: Widgets, Layout, Event Handler, logische Variablen als Referenzen, Zähler-GUI

2.2.: Kompositionalität (vier Zähler-GUIs), Temperaturkonverter-GUI, Tischrechner-GUI, IO-Referenzen, XML als semi-strukturiertes Austauschformat, Darstellung von XML-Dokumenten, Pattern Matching auf XML-Dokumenten, partielle Muster

4.2.: Partielle Muster, ungeordnete Muster, tiefe Muster, negierte Muster, Transformation von XML-Dokumenten
Zusammenfassung