

# Deklarative Programmiersprachen

## WS 2020

Michael Hanus

31. Januar 2021

### Detaillierter Vorlesungsverlauf

- 2.11.: Einführung: Entwicklung von Sprachkonzepten, referenzielle Transparenz, Vorteile im Vergleich zu imperativen Sprachen (Optimierung, Parallelisierung, Zuverlässigkeit, Lesbarkeit), Beispiel Quicksort, Klassifikation von Programmiersprachen, Anwendungen deklarativer Programmiersprachen, Beispiele: Adresssuche in HTML-Seiten, Su Doku-Löser mit Web-Interface
- 3.11.: Funktionale Programmierung: Ausdrücke, Funktionsdefinitionen, Auswertung von Ausdrücken, Redex, strikte und nicht-strikte Sprachen, Fallunterscheidung, bedingte Regeln, Muster, Spezifikationsprache vs. deklarative Programmiersprache, Newtonsches Approximationsverfahren, Gültigkeitsbereich, lokale Deklarationen, Layout-Regel; strenge Typisierung, Datentypen, Basistypen strukturierte Typen (Listen, Zeichenketten, Tupel), funktionale Typen, benutzerdefinierte Datentypen
- 9.11.: *Pattern Matching*: Vorteile von Definitionen mit Pattern Matching, Auswertungsverhalten bei Mustern (Konjunktion, Paralleles Oder), *case*-Ausdrücke, Übersetzung muster-orientierter Funktionsdefinitionen in *case*-Ausdrücke, Übersetzung des *parallel or*, Problem überlappender Regeln, Funktion *diag*, prinzipielles Problem sequentieller Auswertungsstrategien, uniforme Funktionsdefinitionen, Reihenfolgeunabhängigkeit
- 10.11.: *Typsysteme*: streng getypte Sprache, Typ, Typfehler, schwach getypte Sprache, ad-hoc Polymorphismus, parametrischer Polymorphismus, Typinferenz, typkorrekt, Typausdrücke, (Typ-)Substitution, Typinstanz, Typschema, generische Instanz
- 16.11.: Typannahme, Inferenzsystem zur Typprüfung, Typisierung von *twice*, allgemeinsten Typ, Typprüfung für mehrere Funktionen, Typinferenz, Vorgehen, Berechnung allgemeinsten Unifikatoren nach Martelli/Montanari
- 17.11.: Typinferenz für mehrere Funktionen, statischer Aufrufgraph, Grenzen der Typisierung: Selbstanwendung, *funsum*, Polymorphismus 2. Ordnung

*Lazy Evaluation:* Redex, Normalform, LO-Reduktion, schwache Kopfnormalform  
Hamming-Zahlen, Planung mit Prozessnetzen

- 23.11.:** *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, Substitution, Position, Teilterm, Ersetzung, Vergleich von Termpositionen, Reduktionsschritt, Redex
- 24.11.:** kritisches Paar, Kritisches-Paar-Lemma, (schwach) orthogonale Termersetzungssysteme, Reduktionsstrategie (sequenzielle, normalisierende), Reduktionsstrategien LI/LO/PO, linksnormale Termersetzungssysteme
- 30.11.:** Konstruktor, konstruktorbasiertes Termersetzungssystem, definierender Baum, induktiv-sequenziell, Reduktionsstrategie  $\varphi$ , vollständig definierte Funktionen, Eigenschaften von  $\varphi$
- 1.12.:** Motivation zur logisch-funktionalen Programmierung: Verwandtschaftsbeispiel, Aspekte der logisch-funktionalen Programmierung Variablen in initialen Ausdrücken, Extravariablen in Regeln, **failed**, nichtdeterministische Operationen (“?”), Logiksprachen als Spezialfall logisch-funktionaler Sprachen, Spiel 24
- 7.12.:** reguläre Ausdrücke (Darstellung, 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)
- 8.12.:** 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)
- 14.12.:** Strikte Narrowing-Strategien: Innermost Narrowing, Vollständigkeit, Innermost Basic Narrowing, Rückweisung (rejection), Normalisierendes Innermost Narrowing
- 15.12.:** Lazy Narrowing-Strategien: Outermost Narrowing, Lazy Narrowing, Nachteile, Idee von Needed Narrowing, Steuerung von Needed Narrowing durch definierende Bäume, formale Definition von Needed Narrowing, Eigenschaften (Vollständigkeit, Optimalität, Determinismus)
- 4.1.:** Erweiterte definierende Bäume für überlappende Regeln, Definition und Eigenschaften von Weakly Needed Narrowing; Nichtdeterministische Operationen: Realisierung, intuitive Bedeutung, Anwendung (permutation sort), operationale Vorteile, semantische Probleme (call-time choice vs. run-time choice)

- 5.1.:** Residuation: operationale Idee, Vorteile, Anschluss externer Funktionen mittels Residuation, Nachteile von Residuation (Unvollständigkeit, unendliche Berechnungen mit verzögerten Constraints), Kombination von Residuation und Narrowing (flex/rigid branches), Strategie für diese Kombination, Auswertungsstrategie von Curry (optimale definierende Bäume) bedingte Regeln, Funktionen höherer Ordnung (rigides `apply`)
- 11.1.:** Gleichheitstest, strikte Gleichheit, Unterschied `Eq` und `Data`, Typ freier Variablen, Gleichheitsconstraint, Unifikation
- 12.1.:** Gleichheitstest, Gleichheitsconstraint, Modellierung nebenläufiger Objekte (Bankkonto), Erweiterungen: Constraint-Programmierung, CLP(R) (Hypothekenberechnung in Curry), CLP(FD): send-more-money-Beispiel, vordefinierte FD-Constraints, SuDoku-Löser
- 18.1.:** Einkapselte Suche: Motivation, `findall`, Probleme, starke und schwache Einkapselung, Probleme der starken und schwachen Einkapselung, rigide Einkapselung für logische Variablen, set functions, Datentyp `Values` und seine Operationen
- 19.1.:** Beispiel kürzeste Wegesuche 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
- 25.1.:** Ebenenabbildung und stratifizierte Programme, Unifikation “=`<=`” für funktionale Muster, Beispiele `perm`, `descending`, `sort`, `lengthUpToRepeat`, Vereinfachung arithmetischer Ausdrücke
- 26.1.:** GUI-Programmierung: Widgets, Layout, Event Handler, logische Variablen als Referenzen, Zähler-GUI, Kompositionalität von GUIs (vier Zähler-GUIs), Temperaturkonverter-GUI, Tischrechner-GUI, IO-Referenzen
- 1.2.:** Web-Programmierung mit Curry, Analyse von Zugangsprotokollen mit Curry, XML als semi-strukturiertes Austauschformat Darstellung von XML-Dokumenten