Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Einführung in das funktionale Programmieren:

Sommersemester 1997

Übungsblatt Nr. 5

Aufgabe 1

Geben Sie ein Verfahren an mit dem ein beliebiges **KFP** Programm in Haskell übersetzt werden kann.

Aufgabe 2

Einfache Datentypen können wir als Summe von Produkten auffassen, also:

$$\begin{array}{rcl} T & = & T_{1,1} \times \ldots \times T_{1,m_1} \\ & + & T_{2,1} \times \ldots \times T_{2,m_2} \\ & & \vdots \\ & + & T_{n,1} \times \ldots \times T_{n,m_n} \end{array}$$

Geben Sie zu einem solchen Datentyp **KFP**-Funktionen *ohne* Verwendung von Konstruktoren an, die erlauben Elemente des Datentyps darzustellen und darauf zuzugreifen.

Aufgabe 3

Verwenden Sie den Datentyp `KFPexp` vom Blatt 4, um eine Funktion `reduce` in Haskell zu schreiben, die einen **KFP**-Ausdruck um einen Schritt reduziert. Nehmen Sie an ein Argument von `reduce` sei die Funktionsumgebung vom Typ `[(String, [String], KFPexp)]`, in dem für jede Funktionsdefinition ein Eintrag mit dem Namen der Funktion, den Namen der Argumente und dem Ausdruck der rechten Seite der Definition vorliegt.

(wird fortgesetzt)

Aufgabe 4

In der Vorlesung wurden die SKI-Kombinatoren vorgestellt und es wurde angegeben, wie `True` damit dargestellt werden kann. Definieren Sie `And` mit `S`, `K` und `I` und reduzieren Sie `And True True`.