Algorithmen in HASKELL für den 20. Bundeswettbewerb Informatik 2001/2002

Heinrich-Gregor Zirnstein

Kurzfassung der besonderen Lernleistung

Angeregt von meinem Informatiklehrer, Herrn St. Müller, hatte ich in der Jahrgangsstufe 11 die Arbeit an den Aufgaben des 20. Bundeswettbewerbs Informatik aufgenommen, als ich zufällig durch Herrn Dr. Waldmann von der Universität Leipzig auf die mir unbekannte funktionale Programmiersprache HASKELL aufmerksam wurde. Obgleich die mathematisch orientierte funktionale Programmierung hierzulande gegebenenfalls erst in den höheren Semestern erlernt wird, beschloss ich, die Sprache HASKELL ihrer außerordentlichen Ausdrucksstärke wegen im Selbstudium zu erlernen und sogleich bei der Lösung des Bundeswettbewerbs Informatik zu verwenden.

Diese besondere Lernleistung wird in der vorliegenden Arbeit in überschaubarer Form dokumentiert. Dabei beschreibe ich zunächst kurz deren fachpraktische Komponente, meine Teilnahme am Bundeswettbewerbs Informatik: Ich löste die Aufgaben der ersten und zweiten Runde wie gefordert selbstständig, dazu ausnahmslos und als einziger in HASKELL und mit Bravour, so dass ich zur Endrunde nach Frankfurt/Main eingeladen wurde. Dort erkannte mir die Jury nach eingehender Befragung u.a. zur Theorie funktionaler Sprachen schließlich den Bundessieg nebst drei Sonderpreisen zu.

Des Weiteren erläutere ich Grundzüge der Theorie funktionaler Programmierung, indem ich die neuen Konzepte der Formulierung von Algorithmen in funktionalen Sprachen wie HASKELL im Gegensatz zu den herkömmlichen imperativen Sprachen wie PASCAL oder C aufzeige. Der grundlegende Unterschied besteht darin, dass alle Berechnungen in funktionalen Sprachen als mathematische Funktionen aufgefasst werden, deren Resultate nur von den Eingangsdaten abhängen und deren Berechnungsreihenfolge prinzipiell gesehen keinen Unterschied macht. Globale oder sonstige Variablen wie in PASCAL oder C werden untersagt, und die in imperativen Sprachen fundamentale Beachtung der Reihenfolge von Anweisungen wird überflüssig. Auch gibt es keine Programmschleifen mehr, sie werden durch das fundamentalere Konzept der Rekursion vollständig ersetzt. Durch diese Betrachtungsweise können Algorithmen in der Abstraktionsebene formuliert werden, in der sie auch entworfen werden. Die Details der konkreten Steuerung des Computers werden überflüssig und der Programmierer kann sich auf die inhaltlichen Kernpunkte des Algorithmenentwurfs konzentrieren.

Im Hauptteil wird am praktischem Beispiel das Gelernte demonstriert: Exemplarisch präsentiere und bespreche ich meine Lösung zur dritten Aufgabe aus der zweiten Runde des Bundeswettbewerbs Informatik, die meines Erachtens am deutlichsten zeigt, wie vorteilhaft und mit welchen konkreten Konsequenzen sich Algorithmen in HASKELL formulieren lassen. Abschließend füge ich hinzu, dass meine Beschäftigung mit funktionalen Programmiersprachen mit dieser Arbeit natürlich nicht beendet sein wird.

Auf der als Anlage beigefügten Diskette sind auch meine Lösungen in HASKELL aus den ersten beiden Runden des 20. Bundeswettbewerbs Informatik verfügbar, um meine umfangreiche Wettbewerbsleistung nachvollziehbar zu machen. Ebenso wird diese schriftlichen Arbeit hier in elektronischer Form vorgelegt, wobei besonders Wert darauf gelegt wurde, dass der notierte Quelltext direkt von einem HASKELL-Interpreter ausführbar ist.