

Snippet definitions

Fabian Hellauer

February 11, 2016

float def

definition $float = \{real-of-int\ m * 2^{powr\ real-of-int\ e} \mid m\ e.\ True\}$

definition $float' = \{real-of-int\ m * 2^{powr\ real-of-int\ e} \mid m\ e.\ True\}$

float type

IEEE.float

float typeof

real set

float typ

IEEE.float

1 Introduction

The transformation of numerical methods to use machine arithmetic is known to be error-prone. Therefore it is appropriate and important to exercise care to minimize errors. Most of the numerical methods implemented in Isabelle had already been proven to be correct beforehand in Mathematics. While most of these proofs work in a rather general setting, some require additional assumptions or preconditions that are not necessarily given in the context of real operational code, even if it uses the well-known IEEE standard of floating point arithmetic. As a related problem, this standard also introduces a lot of additional cases for operation results with its special values. In consequence, one might consider working with arbitrary precision formats that implement the entire number format using only the infinitely precise integer operations.

2 Fragestellung und Aufgaben

Die Entwicklung des interaktiven Theorembeweisers Isabelle hat das Formalisieren von Beweisen in den letzten Jahren entscheidend vereinfacht. Die Entwickler Nipkow et al. stellen in ihrem "tutorial ..." die Grundfunktionen des Programms vor. Der Umfang der bereits formalisierten Beweise und verfügbaren Module nimmt stetig zu, insbesondere in der großen Bibliothek "Archive of formal proofs" (<http://afp.sourceforge.net/>). Weitere Herausforderungen ergeben sich, wenn die Möglichkeiten für Korrektheitsbeweise auf bisher vernachlässigte Bereiche wie dem Numerischen Programmieren erweitert werden sollen. Kurz gesagt geht in dieser Arbeit darum, die Kluft zwischen den bereits möglichen Beweisen in der Numerik und dem bereits formalisierten IEEE-Standard zu schließen, um in Zukunft Korrektheitsbeweise für numerischen Code formalisieren zu können. In der Arbeit wird dabei ein Ansatz eines Dateiformats unter Ausnutzung der floating point Operationen gewählt. Beschrieben wird er unter anderem in (Shewchuck et al.). Der Titel der Arbeit "Multiple Precision Floats in Isabelle/HOL" bringt das für Kenner dieser Idee auf den Punkt. Voraussetzung für ein tieferes Verständnis der Aufgabenstellung ist das Verständnis der Beweissprache Isabelle. Hilfreich war dafür neben den ausführlichen Gesprächen mit meinem Betreuer der Besuch des Kurses "Semantics of Programming languages". Desweiteren war die Beschäftigung mit der bereits vorhandenen Formalisierung des IEEE-Standards hilfreich, da diese schon die Erstellung von Hardware-Operationen benutzendem ML-Code ermöglicht. Aus diesen Ausgangsbearlegungen ergaben sich eine Reihe konkreter Aufgaben für die gewünschte Bereitsstellung des Datenformats: Erstellung des Codes für die Umwandlung aus anderen Formaten, Sicherstellung der Übersetzbarkeit in Hardware-Operationen benutzenden Code, Beweise zur Korrektheit unter gewissen Annahmen, eine sinnvolle Überset-

zung vom in den Verffentlichungen beschriebenen imperativen Code fr Operationen in den funktionalen Stil von Isabelle/HOL ohne Verschlechterung der Laufzeitkomplexitt (Herausforderung), Bereitsstellung von mglichst vielen Lemmas ber die Eigenschaften des Codes, Erstellung eines vorlufigen Grundgersts fr vollstndige Korrektheitsbeise in der Zukunft und das Testen des generierten Codes auf Korrektheit und Laufzeit. Fr die Attraktivitt von Isabelle fr numerische Lsungen ist der Aufbau einer groen Sammlung von solchen Modulen mit Ausnutzung der Hardware - wie es das "MPF"-Format in der Arbeit anstrebt - von zentraler Bedeutung.

Die Korrektheisbeweise der TwoSum-Methode, die komplizierte arithmetische Argumentationen in der Darstellung erfordern, war bewusst nicht Teil der Aufgabenstellung.