Institut für Theoretische Informatik
Logik und Formale Methoden
Prof. Dr. P. H. Schmitt, C. Engel, D. Faragó, R. Krenický, M. Ulbrich, B. Weiß

18.01.2010

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 8: KeY

Aufgabe 20 — Interpretieren offener Beweise

Wenn eine Methode nicht ihre Spezifikation erfüllt und die entsprechende Beweisverpflichtung folglich auch nicht bewiesen werden kann, dann können aus dem resultierenden Beweisbaum nützliche Informationen zum Finden des Fehlers gewonnen werden. Relevant sind hier in erster Linie die offenen Äste des Beweisbaums, aus denen sich herauslesen läßt, in welchen Fällen und bei Ausführung welcher Ausführungspfade im betrachteten Code ein Fehler auftritt, d.h. die Spezifikation verletzt wird.

Betrachten wir hierzu ein aktuelles Beispiel: Am 31. Dezember 2008 fielen weltweit alle Zune-Geräte der ersten Generation aus. Die Ursache war ein interner Fehler bei der Handhabung von Schaltjahren wie hier nachzulesen ist:
`http://www.heise.de/newsticker/meldung/Zune-Ausfall-Fremder-Code-als-Ursache-193332.html`

Eine Java-Portierung (der auf heise.de verlinkte Code ist in C geschrieben) des dafür angeblich verantwortlichen Codes finden Sie in der auf der Praktikumswebseite verfügbaren Datei `Zune2K9.tgz`. Die Methode `RTC.convertDays` rechnet Anzahl der Tage, die seit dem 01.01.1980 vergangen sind, in Jahre, Monate und Tage um. Der Zune-Bug rührt nun daher, daß diese Methode unter bestimmten Bedingungen, die in dieser Aufgabe mit KeY reproduziert werden sollen, nicht terminiert.

Zur Reproduktion des Bugs gehen Sie folgendermaßen vor:

- Entpacken Sie Zune2K9.tgz entpacken und öffnen Sie das Verzeichnis Zune2K9 mit KeY
- Im sich dann öffnenden Proof Obligation BrowserRTC.convertDays und die Beweisverpflichtung EnsuresPost auswählen
- Sicherstellen, daß folgende Strategieoptionen bei der Strategie *Java DL* eingestellt sind:
  - Goal Chooser: Depth First
  - Stop at: Default
  - Loop treatment: Invariant
  - Method t.: Contracts
  - Query t.: Expand
  - Arithmetic t.: DefOps
  - Quantifier t.: No Splits with Progs
- Max. Rule Applications: 2000 (oder mehr)

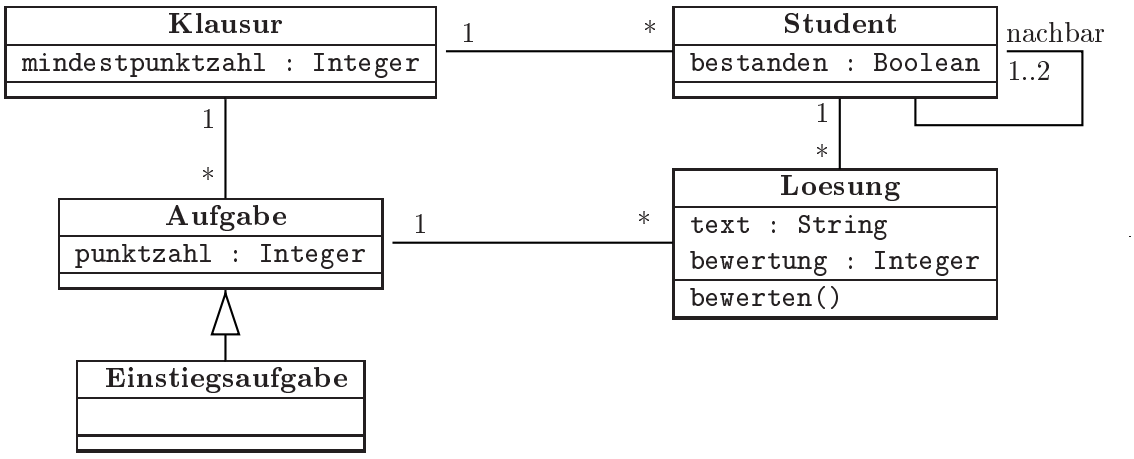
Drückt man jetzt den grünen Knopf, bleibt ein Beweisziel offen. Interpretieren Sie dieses Beweisziel und den dazugehörigen Ast im Beweisbaum. Wie läßt sich der Fehler mit Hilfe

des offenen Beweises finden? Tipp: Die Option *Hide Intermediate Proofsteps* im Kontextmenü kann hierbei hilfreich sein, um sich einen Überblick über die Struktur des Beweises und des offenen Beweisastes zu verschaffen.

Korrigieren Sie den Fehler und versuchen Sie erneut den Beweis durchzuführen.

Aufgabe 21 — Spezifizieren und Verifizieren

Gegeben sei folgendes bereits in Aufgabe 7 auf Übungsblatt 3 vorgestellte Programm:



Eine bereits mit JML-Spezifikationen versehene Version dieses Programms ist in der Datei `Klausuren2.java` auf der Praktikumswebseite verfügbar.

Beweisen Sie die Korrektheit der Methodenspezifikation von `Student.macheLoesungen` (Beweisverpflichtung: *EnsuresPost*). Hierzu müssen sie die Invariante, Variante und Assignable-Klausel der in `macheLoesungen` enthaltenen Schleife spezifizieren. Außerdem ist zu beachten, daß im *Contract Configurator* im Tab *Assumed Invariants Select All* gewählt wurde. Die Strategieoptionen können von der vorherigen Aufgabe übernommen werden.

Spezifizieren Sie außerdem

- einen `exceptional_behavior` Vertrag, der beschreibt, wann eine `IllegalArgumentException` geworfen wird und das es für diesen Fall nur zu einer Exception dieses Typs kommen kann.
- einen `normal_behavior` Vertrag, der den Fall abdeckt, daß die Lösungen mit denen eines Nachbarn übereinstimmen.

Beweisen Sie auch diese beiden Verträge (Beweisverpflichtung: *EnsuresPost*).

Abgabe bis Mittwoch, 27.01.
Die Abgabe erfolgt über die Praktikums-Webseite. Es braucht pro Gruppe und Aufgabe nur *eine* Lösung abgegeben werden. Bitte dokumentieren Sie Ihre Lösungen ausreichend und seien Sie darauf vorbereitet, sie auf Nachfrage zu erklären.

Praktikums-Webseite: `http://lfm.iti.kit.edu/keyprakt0910.php`

Christian Engel:	R. 106,	Tel. 608-4338,	E-Mail: `engelc@ira.uka.de`
David Faragó:	R. 308,	Tel. 608-7322,	E-Mail: `farago@ira.uka.de`
Roman Krenický:			E-Mail: `krenicky@ira.uka.de`
Mattias Ulbrich:	R. 106,	Tel. 608-4338,	E-Mail: `mulbrich@ira.uka.de`
Benjamin Weiß:	R. 309,	Tel. 608-4324,	E-Mail: `bweiss@ira.uka.de`