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

31.01.2011

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 8: KeY

Aufgabe 19 — Verifikation im Praktikumsprojekt

Sie haben eine Version Ihres Bankenprojekts zugesandt bekommen oder werden es noch bekommen. Im Wesentlichen haben wir GUI- und Test-Klassen entfernt, `System.out.println` Anweisungen auskommentiert, und evtl. kleine Syntaxfehler in der Spezifikation behoben. Insbesondere haben wir keine Fehler in Implementierung oder Spezifikation eingebaut, die nicht schon vorher vorhanden waren.

Führen Sie die im Folgenden beschriebenen Verifikationsschritte mit KeY durch. Verwenden Sie dabei die in Aufgabe 18 auf Blatt 7 angegebenen Strategieoptionen. Sollten Sie feststellen, dass ein Beweis nicht schließbar ist, kann der "Fehler" in der Spezifikation, der Implementierung oder in beidem liegen. Die Ursache sollte dann beseitigt und die Verifikation erfolgreich abgeschlossen werden. Nähere Hinweise finden Sie in den einzelnen Aufgabenbeschreibungen unten.

(a) Methode `UnboundedPermanentAccountArray::get(int)`. Beweisen Sie für Ihre beiden Verträge jeweils die "EnsuresPost"-Beweisverpflichtung.

(b) Methode `UnboundedPermanentAccountArray::enlargeArray()`. Beweisen Sie für den Vertrag die "EnsuresPost"-Beweisverpflichtung.

- Überarbeiten Sie Ihren Vertrag zunächst nocheinmal: Richtlinie ist, dass der Vertrag genau die Informationen enthalten sollte, die für einen Aufrufer der Methode `enlargeArray` relevant sind. Im Moment enthält der Vertrag zum einen Informationen, die eigentlich unwichtig sind, und zum anderen fehlen wichtige Informationen.
- Wie üblich müssen Sie für die Behandlung der Schleife eine passende Schleifeninvariante, -variante und `assignable`-Klausel angeben.

(c) Methode `UnboundedPermanentAccountArray::push()`. Beweisen Sie für den Vertrag die "EnsuresPost"-Beweisverpflichtung.

- Verstärken Sie auch diesen Vertrag zunächst, ähnlich wie bei `enlargeArray`.
- Bei der Verifikation wird der Vertrag für `enlargeArray` benutzt (kontrolliert durch die Strategieeinstellung "Method treatment: Contracts").

(d) Methode `CentralHost::getAccount(int)`. Beweisen Sie für Ihre beiden Verträge die "EnsuresPost"-Beweisverpflichtung.

- Die Korrektheit der Methode hängt von Klasseninvarianten anderer Klassen ab. Identifizieren Sie diese Klassen und aktivieren Sie die nötigen Invarianten beim Starten der Beweise im "Contract Configurator" unter "assumed invariants", um sie für die Verifikation verwenden zu können.

(e) Fügen Sie dem Programm eine neue (leere) Unterklasse von `PermanentAccount` hinzu, und versuchen Sie erneut, den Vertrag für `UnboundedPermanentAccountArray::push()` zu beweisen.
Welches Problem tritt auf, und warum? Können Sie die Spezifikation verstärken, so dass der Beweis wieder gelingt?

Abgabe bis Freitag, 11.02.
Abgabe (als Java-, ASCII- oder PDF-Dateien, oder tar-Archiv solcher) per E-Mail an Benjamin Weiß. 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/keyprakt1011.php`

David Faragó:	R. 308,	Tel. 608-47322,	E-Mail: `farago@ira.uka.de`
Christoph Scheben:	R. 106,	Tel. 608-44338,	E-Mail: `scheben@ira.uka.de`
Mattias Ulbrich:	R. 106,	Tel. 608-44338,	E-Mail: `mulbrich@ira.uka.de`
Benjamin Weiß:	R. 309,	Tel. 608-44324,	E-Mail: `bweiss@ira.uka.de`