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

09.12.2009

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 5: ESC/Java2

Aufgabe 13 — Einführung in ESC/Java2

ESC/Java2 ist ein Werkzeug, das die Korrektheit von Java-Programmen bzgl. einer JML-Spezifikation überprüfen kann. Im Unterschied zum Testen mit RAC oder JMLUnit führt ESC/Java2 eine *statische* Verifikation durch, d.h., der Quellcode wird untersucht, ohne das Programm tatsächlich auszuführen. Vorteile dieses Ansatzes sind vor allem, dass keine Testfälle geschrieben werden müssen, und dass die Prüfung *alle* möglichen Eingabewerte und alle Ausführungspfade durch das Programm berücksichtigen kann, nicht nur diejenigen, die durch Testfälle abgedeckt sind.

Lesen Sie sich die auf der Webseite verfügbaren Folien zu ESC/Java2 aus dem Praktikum vom letzten Jahr durch. Nehmen Sie dabei insbesondere folgende Punkte zur Kenntnis:

- *Modularität* der Verifikation: ESC/Java2 prüft jede Methode einzeln. Ruft eine Methode eine andere auf, wird zur Behandlung dieses Methodenaufrufs nur die *Spezifikation* der aufgerufenen Methode benutzt, niemals ihre Implementierung. Es ist Aufgabe des Benutzers, dafür zu sorgen, dass diese Spezifikation aussagekräftig genug ist, um die Korrektheit der aufrufenden Methode daraus abzuleiten.

- Fehlende *Vollständigkeit*: Es kommt vor, dass ein eigentlich korrektes Programm von ESC/Java2 als fehlerhaft beanstandet wird, weil das Werkzeug nicht in der Lage ist, die Korrektheit zu beweisen. (Da das Problem der Programmverifikation unentscheidbar ist, muss es solche Fälle schon aus theoretischer Sicht geben.)

- Fehlende *Korrektheit*: Um die Benutzbarkeit möglichst einfach zu halten, macht ESC/Java2 auch bei der Korrektheit Kompromisse. Es gibt also fehlerhafte Programme, die das Werkzeug fälschlich als korrekt klassifiziert.

- Voreinstellung für `non_null`/`nullable`: In JML ist seit einiger Zeit "`non_null`" die Voreinstellung, aber ESC/Java2 hat diesen Wandel nicht ganz nachvollzogen. In der von uns verwendeten Version 2.0.5 mit Option `-nonNullByDefault` scheint `non_null` zwar als Voreinstellung für Methodenparameter und -rückgabewerte verwendet zu werden, nicht aber für Felder. Geben Sie im Zweifelsfall alle `non_null`/`nullable`-Annotationen explizit an.

Sie können ESC/Java2 mit einem Befehl wie "`escj MyClass.java`" auf eine Klasse anwenden. Mit Hilfe der Option `-routine` können Sie auch nur einzelne Methoden prüfen.

Aufgabe 14 — Bag und Amount

Laden Sie von der Praktikums-Webseite die Dateien `Bag.java` und `Amount.java` herunter und lösen Sie die in den Kommentaren beschriebenen Aufgaben. Ziel ist es, den Quellcode mit JML-Spezifikationen zu annotieren und eventuell vorhandene Bugs zu beseitigen, bis ESC/Java2 schließlich keine Warnungen mehr ausgibt. Bei `Amount.java` sind vorher noch einige in natürlicher Sprache gegebene Eigenschaften als JML-Klasseninvarianten zu formalisieren.

In beiden Fällen soll die fertige Spezifikation keine `assume`-Anweisungen enthalten. Verwenden Sie auch keine Optionen wie `NoWarn`, mit denen Warnungen unterdrückt werden.

Zum Schluß: Glauben Sie, dass Sie *alle* Probleme gefunden haben? Wie sicher sind Sie sich, dass der Code jetzt korrekt ist? Haben Sie Ideen, wie JML oder ESC/Java2 verbessert werden könnten?

Aufgabe 15 — Verifizierte Authentifizierung

Wenden Sie ESC/Java2 auf Ihre Implementierung des Authentifizierungssystems von Aufgabe 8 an. Gibt das Werkzeug Warnungen aus? Wenn ja, erklären Sie den Grund für die Warnungen, und versuchen Sie, alle Warnungen zu beseitigen, indem Sie zusätzliche Spezifikationen hinzufügen und eventuelle Bugs beheben. Dokumentieren Sie Ihr Vorgehen.

Abgabe bis Mittwoch, 16.12.
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`