Karlsruher Institut für Technologie
Institut für Theoretische Informatik
Prof. Dr. Bernhard Beckert
Dr. Thorsten Bormer, Dr. Vladimir Klebanov, Dr. Mattias Ulbrich

Formale Systeme, WS 2015/2016

Praxisaufgabe 2: Theorembeweiser

Abgabe der Lösung bis zum 29.02.2016 über die ILIAS-Seite zur Vorlesung
`https://ilias.studium.kit.edu/goto_produktiv_crs_472063.html`

Hinweis: Bitte beachten Sie, dass die Praxisaufgabe in Einzelarbeit und selbständig zu bearbeiten ist. Wir behalten uns vor, die selbständige Bearbeitung dadurch stichprobenartig zu prüfen, dass wir uns die eingereichte Lösung erklären lassen.

Für die vollständige Lösung dieser Praxisaufgabe erhalten Sie **10 Übungspunkte**. Bitte beachten Sie die Erläuterung zu Übungspunkten auf der Webseite zur Vorlesung. Für unvollständige Lösung werden die Übungspunkte anteilig vergeben.

A Informationen zum KeY-System

Was ist KeY? Zusammen mit unseren Partnern, unter anderem an der Technische Universität Darmstadt, wird an unserem Institut das KeY-System entwickelt. Es ist ein Softwareverifikationswerkzeug, mit dem die Übereinstimmung von Java Card-Software und ihrer Spezifikation formal bewiesen werden kann.

Die Logik, auf der das KeY-System basiert, ist eine sortierte dynamische Prädikatenlogik. In dieser dynamischen Logik ist die in der Vorlesung eingeführte Prädikatenlogik vollständig enthalten. Deshalb können wir KeY auch benutzen, um rein prädikatenlogische Probleme zu formulieren und zu beweisen.

Literatur zu KeY Auf der Internetseite der Vorlesung steht eine Einführung zu Beweisen von prädikatenlogischen Formeln mit KeY (`KeYIntro.pdf`). Bitte arbeiten Sie diese Einführung durch.

Für weitergehende Fragen bietet sich die Lektüre des KeY-Buches [BHS07] an und darin vor allem Kapitel 10, das eine tiefere Einführung in KeY bietet. Kapitel 2 des Buches erklärt fundiert die prädikatenlogischen Grundlagen, auf denen KeY basiert.

Installation Das KeY-System besitzt eine graphische Benutzeroberfläche und ist komplett in Java geschrieben, so dass es auf jeder Plattform, für die eine virtuelle Maschine für Java zur Verfügung steht, lauffähig ist.

Auf der Webseite zu dieser Vorlesung können Sie die Version von KeY finden, die Sie zur Lösung dieses Übungsblattes verwenden sollen. Wenn Sie die Software „Java Web Start“ installiert haben (auf fast allen modernen Systemen mit Java der Fall), können Sie KeY direkt aus dem Internetbrowser heraus starten.

Bei neueren Java-Versionen sind die Sicherheitseinschränkungen verschärft worden. Es kann daher sein, dass Sie das Programm nicht über Web Start starten können. Wir stellen auf der Vorlesungswebseite daher auch eine herunterladbare Version von KeY zur Verfügung.

Bitte verwenden Sie in jedem Fall die KeY-Version von der Vorlesungshomepage und nicht die Versionen auf den Seiten des Tools.

Hinweis zur Konfiguration von KeY Für die Aufgaben dieses Blattes empfiehlt es sich, die Standardeinstellungen des KeY-Systemes zu belassen, wie sie zum Zeitpunkt des Systemstarts sind.

B Aufgabe: Spezifikation und Verifikation eines Java-Programms

In dieser Aufgabe soll das gewünschte Verhalten eines kleinen Java-Programms in JML formalisiert werden. Und es soll mit KeY bewiesen werden, dass das Programm sich tatsächlich so verhält.

Gegeben sei die Implementierung der Methode `isSubset`, die als Parameter zwei `int`-Arrays `a` und `b` übergeben bekommt und *genau dann* `true` zurückliefert, wenn alle Elemente des Arrays `a` auch Elemente des Arrays `b` sind:

```
class SubsetArray {

  /*@ public normal_behaviour
    @ assignable \nothing;
    @ ensures ...;
    @*/
    public static boolean isSubset(int[] a, int[] b) {

       /*@ loop_invariant ...;
         @ assignable \nothing;
         @ decreases ...;
         @*/
        for (int i=0; i < a.length; i++) {
            boolean found = false;

          /*@ loop_invariant ...;
            @ assignable \nothing;
            @ decreases ...;
            @*/
            for (int j=0; (j < b.length) && !found; j++) {
                if (a[i] == b[j]) found = true;
            }

            if (!found) return false;
        }

        return true;
    }
}
```

Aufgabe

(a) Spezifizieren Sie das gewünschte Verhalten der Methode, indem Sie den vorgegebenen Methodenvertrag in JML vervollständigen.

(b) Beweisen Sie die Korrektheit der Implementierung gegenüber dem Methodenvertrag. Vervollständigen Sie dazu die Schleifeninvarianten und -varianten (`decreases`-Klauseln).

(c) Speichern Sie den geschlossenen Beweis in einer Datei mit dem Namen `isSubset.key.proof` ab.

Abgabe Bitte laden Sie die Datei `isSubset.key.proof` mit Ihrem Beweis **und** die von Ihnen vervollständigte Datei `SubsetArray.java` mit dem annotierten Programm mittels der dafür vorgesehenen Formularfelder in ILIAS hoch.

Weitere Hinweise Sollte die Implementierung und Ihre Spezifikation konsistent sein, und die notwendigen Zusatzannotationen hinreichend, so funktioniert der Beweis in der Regel automatisch (z.B. durch Betätigen des grünen „Play"-Knopfes).

Erfahrungsgemäß klappt Beweisen aber nicht auf Anhieb, da die Spezifikation und die Zusatzannotationen Fehler enthalten können. Zum „Debuggen" empfiehlt es sich, den „Beweis-Autopiloten" zu nutzen. Wählen Sie dafür nach Laden der Beweisverpflichtung den Punkt „Full Auto Pilot" aus dem Kontextmenü „Strategy Macros", das in der Sequenzen- oder der Beweisbaumansicht verfügbar ist. Der Autopilot strukturiert den Beweis stärker anhand der zu beweisenden Aussage: jeder Einzelteil der zu beweisenden Aussage wird als separates Beweisziel behandelt. An den offenen Zielen kann man in diesem Fall besser erkennen, welcher Teil der Korrektheitsaussage nicht (automatisch) bewiesen werden konnte.

KeY wird mit einigen Beispielen ausgeliefert, diese können unter „Load Example" im File-Menü aufgerufen werden. Als Einstieg sind die Beispiele „Sum and Max", „Binary Search", und „Remove Duplicates" gut geeignet.

Literatur

[BHS07] Bernhard Beckert, Reiner Hähnle, and Peter H. Schmitt, editors. *Verification of Object-Oriented Software: The KeY Approach.* LNCS 4334. Springer-Verlag, 2007. Verfügbar im KIT-Netz unter `http://link.springer.com/book/10.1007/978-3-540-69061-0`.