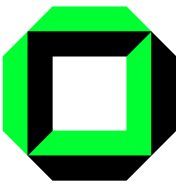

Universität Karlsruhe
Institut für Theoretische Informatik
Prof. Dr. P. H. Schmitt
Dr. T. Käufl, C. Engel, B. Weiß

12. Dezember 2007

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 7

Aufgabe 1

Passen Sie in Ihrer Endversion der Datei `Bag.java` aus Blatt 5, Aufgabe 2, die Methode `removeAll` an, so dass sie folgendermaßen aussieht:

```
/*@ normal_behavior
  @   ensures (\forall int x; 0 <= x && x < n; contents[x] != elt);
  @*/
 void removeAll(int elt) {
     for (int i = 0; i < n; i++) {
         if (contents[i] == elt ) {
             n--;
             contents[i] = contents[n];
             i--;
         }
     }
 }
```

Beweisen Sie mit KeY den "normal_behavior speccase" für diese Methode.

Hinweise:

- Ihre Hauptaufgabe ist, für die Schleife eine geeignete Invariante, `assignable`-Klausel und Variante (`decreases`-Klausel) anzugeben. Fügen Sie diese als JML-Annotationen in den Code ein.
- Sie sollten in Ihrer Datei bereits Klasseninvarianten spezifiziert haben, die besagen, dass `contents` nicht `null` ist, und dass sich `n` im Intervall [0, `contents.length`] bewegt. Diese Klasseninvarianten erscheinen automatisch in der Vorbedingung der Beweisverpflichtung. Sie sind nötig, um den Beweis schließen zu können. Die Aussage über den Wertebereich von `n` müssen Sie auch in die *Schleifen*invariante aufnehmen.

  Achtung: KeY kennt die Abkürzung `/*@ non_null *@/` nicht. Falls Sie diese verwendet haben, formulieren Sie die Klasseninvariante entsprechend um.
- Wenn Sie alles richtig gemacht haben, kann der Beweis vollautomatisch durchgeführt werden. Aktivieren Sie dazu im Reiter "proof search strategy" die Strategie "Java DL" und nehmen Sie folgende Einstellungen vor:
  - "Loop treatment": "Invariant" (Schleifen sollen mit der Invariantenregel behandelt werden)
  - "Method treatment" - "Expand" (Methoden sollen durch Auspacken des Rumpfes behandelt werden)

– "Quantifier treatment - No Splits with Progs" (Quantoren sollen heuristisch instantiiert werden, aber der Beweis soll dadurch nicht aufgespaltet werden, solange noch Programme enthalten sind)

Wenn Sie nun die Strategie starten, sollte der Beweis nach größenordnungsmäßig 1000 Regelanwendungen geschlossen werden.

- Schauen Sie sich den erfolgreichen Beweis an und verschaffen Sie sich ein grundlegendes Verständnis seiner Struktur. Insbesondere sind die anfängliche Beweisverpflichtung und die von der Invariantenregel erzeugten Beweisknoten interessant.
- Geben Sie sowohl den gespeicherten Beweis als auch die veränderte Datei `Bag.java` ab.

Aufgabe 2

Fügen Sie in der Endversion Ihrer Datei `Amount.java` aus Blatt 5, Aufgabe 2, folgenden Vertrag für die Methode `subtract` ein:

```
/*@ public normal_behavior
  @   requires a != null;
  @   assignable \nothing;
  @   ensures \result != null;
  @   ensures \result.euros*100+\result.cents
  @              == euros*100+cents - (a.euros*100+a.cents);
  @*/
  public Amount subtract(Amount a){
      ...
```

Ihre Aufgabe ist auch dieses Mal, den "normal_behavior speccase" mit KeY zu verifizieren. Dabei sollen die Aufrufe der Methoden `negate` und `add` *nicht* durch Auspacken der Methodenrümpfe (Regel `methodBodyExpand`), sondern durch Verwenden von Verträgen für die beiden aufgerufenen Methoden behandelt werden.

Hinweise:

- Die Verträge für `negate` und `add` müssen Sie zunächst schreiben. Orientieren Sie sich dabei am vorgegebenen Vertrag für `subtract`. Die beiden Verträge sollen korrekt sein (d.h. von der jeweiligen Methodenimplementierung erfüllt werden), Sie brauchen das aber nicht zu beweisen.
- Um den Beweis durchzuführen, gehen Sie am besten folgendermaßen vor: Wählen Sie in den Strategieoptionen für "Method treatment" die Einstellung "none". Lassen Sie dann die Strategie laufen, die nun jedesmal anhält, wenn sie einen Methodenaufruf erreicht. Der erste Aufruf ist der von `subtract` selbst; packen Sie hier den Rumpf aus, indem Sie die Regel `methodBodyExpand` anwenden. Die Aufrufe von `negate` und `add` behandeln Sie dagegen mit der Regel "Use Method Contract". Bei Anwendung dieser Regel öffnet sich ein Dialogfenster, in dem Sie aus den existierenden Verträgen für die aufgerufene Methode einen auswählen können (wahrscheinlich gibt es nur einen). Der Beweis sollte ansonsten automatisch zugehen, nach insgesamt größenordnungsmäßig 400 Schritten.
- Geben Sie sowohl den gespeicherten Beweis als auch die veränderte Datei `Amount.java` ab.

Abgabe bis 09.01.

Es braucht pro Gruppe nur *eine* Lösung abgegeben werden.
Die Abgabe der Übungsblätter erfolgt mit dem SVN System. Dazu legen Sie die abzugebenden Dateien im SVN ab und kopieren sie mit SVN in den Unterordner *abgabe/<nr>* wie in Aufgabe 2 auf Blatt 1 beschrieben.
Einige Aufgaben verlangen eine schriftliche Bearbeitung, diese ist dann je nach Komplexität als ASCII, html, ps- oder pdf-Dokument abzugeben. Auf *keinen* Fall im MS Word `doc`-Format.

Praktikums-Webseite: `http://i12www.ira.uka.de/~engelc/lehre/keypraktWS0708/`

Dr. Thomas Käufl: Zi. 207, Tel. 608-6286, E-Mail: `kaeufl@ira.uka.de`
Christian Engel: Zi. 106, Tel. 608-4338, E-Mail: `engelc@ira.uka.de`
Benjamin Weiß: Zi. 309, Tel. 608-4324, E-Mail: `bweiss@ira.uka.de`