Algorithmen und Datenstrukturen

Dr. Martin Moritz Kleine

Aufgabenblatt 2 : Spezifikation und Verifikation mit Dafny

Im Folgenden sind Algorithmen gegeben, für die Sie geeignete Verträge bestimmen sollen, um diese dann mit Dafny[1] zu verifizieren. Als Richtline für Verträge gilt, dass die Vorbedingungen (*requires*) so allgemein wie möglich sein sollen, während die Zusicherungen (*ensures*) so speziell wie möglich sein sollen. Insbesondere das *binarySearch* Beispiel des Dafny Tutorials sollten Sie sich zur Einarbeitung anschauen.

Die Abgabe erfolgt als Einsendung eines Permalinks auf *rise4fun.com*.

Verwenden Sie http://rise4fun.com/Dafny/l1t als Vorlage.

Übungsaufgabe 2.1: *Die Fibonacci Zahlen*

Implementieren und verifizieren Sie die rekursive Version von *Compute_Fib* aus oben genannter Vorlage (entsprechend *fib3* des vorherigen Aufgabenzettels).

Übungsaufgabe 2.2: *Lösen von Rekurrenz-Gleichungen*

Gegeben sei die folgende Rekurrenz:

$$f(n) = \begin{cases} 2, & n = 0 \\ 3 \cdot f(n-1) + 2, & n > 0 \end{cases}$$

1. Geben Sie einen geschlossenen Ausdruck für $f(n)$ an.
2. Implementieren Sie f als Dafny Methode und geben Sie die von Ihnen bestimmte geschlossene Form als Vertrag an. Ergänzen Sie die nötigen Invarianten oder Assertions so dass die Methode verifiziert werden kann.

Übungsaufgabe 2.3: *Absichern von Laufzeitabschätzungen*

Bei Laufzeitabschätzung von Algorithmen geht man davon aus, dass alle "eingebauten" Konstrukte (wie Addition, Zuweisung usw.) eine konstante Zeit benötigen. Die Konstante selbst kennen wir nicht, sie ist typischerweise auch stark von der Hardware abhängig. Die Laufzeit von Prozeduren hängt dagegen i.a. von den Argumenten ab.

1. Finden Sie heraus, was die folgenden Algorithmen berechnen und überlegen Sie sich geeignete Verträge dafür.
2. Spezifizieren (verwenden Sie also die in der vorherigen Teilaufgabe entwickelten Verträge) und implementieren Sie die folgenden Algorithmen als Dafny Methoden. Ergänzen Sie die nötigen Invarianten oder Assertions so dass die Methoden verifiziert werden können.
3. Bestimmen Sie Anzahl der Operationen, die jeder der Algorithmen ausführt.

[1] Project Homepage: http://dafny.codeplex.com/, Tutorial: http://rise4fun.com/Dafny/tutorial/guide

4. Erweitern Sie den Dafny Code für jede der Methoden um eine Ghost-Variable *counter* und sichern Sie mittels einer *assertion* zu, dass die von Ihnen bestimmte Anzahl der Operationen stimmt.

Algorithm 1

```
1: x = 0
2: for i = 1 to n do
3:     x = x + A[i]
4: end for
5: return x
```

Algorithm 2

```
1: r = 1
2: for i = 1 to k do
3:     r = r * x
4: end for
5: return r
```

Algorithm 3

```
1: for i = 1 to n do
2:     A[i] = i
3: end for
4: for i = 1 to n do
5:     C[i] = 0
6:     for j = n downto 1 do
7:         if A[j] > C[i] then
8:             C[i] = A[j]
9:         end if
10:    end for
11: end for
12: return C
```