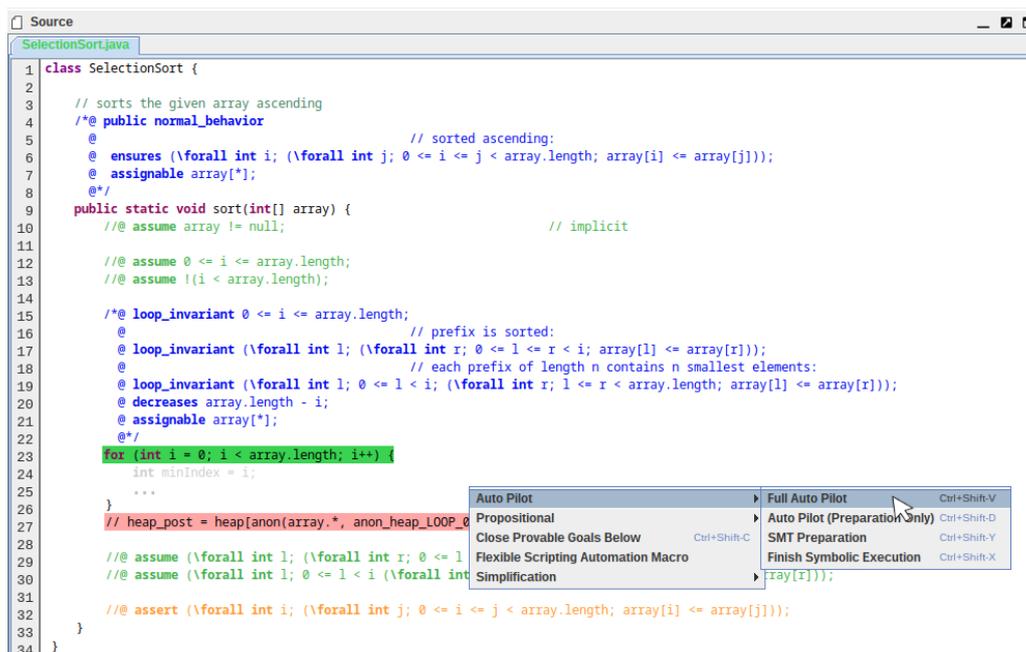


Masterarbeit / Praxis der Forschung

Neue Interaktionsideen für KeY

Hintergrund. KeY ist ein am Lehrstuhl entwickeltes Tool zur deduktiven Verifikation von Java-Programmen. Es unterstützt sowohl automatische als auch interaktive Beweissuche und arbeitet mit *Java Dynamic Logic (JavaDL)* und dem Sequenzkalkül.

Bei der Codierung des Problems als JavaDL-Formel und den anschließenden Regelanwendungen entstehen allerdings einige Terme, die zwar technisch notwendig sind, aber die Lesbarkeit für Benutzer stark einschränken (z.B., dass die in einer Formel vorkommenden Heap-Ausdrücke auch erlaubte Java-Heaps sind).



```

1 class SelectionSort {
2
3 // sorts the given array ascending
4 /*@ public normal_behavior
5 @
6 @ ensures (\forallall int i; (\forallall int j; 0 <= i <= j < array.length; array[i] <= array[j]));
7 @ assignable array[*];
8 @*/
9 public static void sort(int[] array) {
10 //@ assume array != null; // implicit
11
12 //@ assume 0 <= i <= array.length;
13 //@ assume !(i < array.length);
14
15 /*@ loop_invariant 0 <= i <= array.length;
16 @ // prefix is sorted:
17 @ loop_invariant (\forallall int l; (\forallall int r; 0 <= l <= r < i; array[l] <= array[r]));
18 @ // each prefix of length n contains n smallest elements:
19 @ loop_invariant (\forallall int l; 0 <= l < i; (\forallall int r; l <= r < array.length; array[l] <= array[r]));
20 @ decreases array.length - i;
21 @ assignable array[*];
22 @*/
23 for (int i = 0; i < array.length; i++)
24 int minIndex = i;
25 ...
26 }
27 // heap_post = heap[anon(array.*; anon_heap_LOOP_0
28 //@ assume (\forallall int l; (\forallall int r; 0 <= l <= r < array.length; array[l] <= array[r]));
29 //@ assume (\forallall int l; 0 <= l < i (\forallall int
30
31
32 //@ assert (\forallall int i; (\forallall int j; 0 <= i <= j < array.length; array[i] <= array[j]));
33
34 }

```

Vision für die Interaktion auf Source-Code-Ebene: Eine Ansicht für Java, JML **und** Beweiszustand.

Aufgabe.

Ziel der Arbeit ist es, eine neue Ansicht zu entwickeln und zu implementieren, in der sowohl der Java-Quellcode, die ursprüngliche JML-Spezifikation als auch der aktuelle Beweiszustand menschenlesbar dargestellt werden. Außerdem soll untersucht werden, wie Interaktion auf dieser neuen Ansicht möglich ist.

Ihr Profil. Sie sollten solide Java-Kenntnisse besitzen sowie das Modul *Formale Systeme* erfolgreich abgeschlossen haben.

Kontakt

Wolfram Pfeifer

wolfram.pfeifer@kit.edu

50.34 R228