

# Übungszettel 1

## Aufgabe 1: Petersons Algorithm

In der Vorlesung wurde euch *Petersons Algorithmus* zum gegenseitigen Ausschluss von zwei auf einen kritischen Abschnitt zugreifenden Prozessen vorgestellt. Dieser Algorithmus ist in Listing 1 abgedruckt. Hierbei sind beide Prozesse über eine Eindeutige Prozess-ID aus  $\{0, 1\}$  identifiziert. Beide Prozesse rufen vor dem Betreten des kritischen Abschnitts `enter_region()` und nach dem Verlassen `leave_region()` mit ihrer jeweiligen Prozess-ID auf.

Listing 1: Petersons Algorithmus

```
1 volatile int turn;
2 volatile bool interested[2] = {false, false};
3
4 /* pid ist aus {0,1} */
5 void enter_region(int pid) {
6     interested[pid]= true;
7     turn = pid;
8     while (turn == pid && interested[1-pid]);
9 }
10
11 void leave_region(int pid) {
12     interested[pid] = false;
13 }
```

- a) Erläutert die Funktionsweise dieses Algorithmus.
- b) Modelliert den Algorithmus mit dem UPPAAL tool.
- c) Stellt die nötigen Anforderungen für die Korrektheit des Algorithmusses auf und prüft mit UPPAAL das Modell gegen die Anforderungen.
- d) Ist der Gegenseitige Ausschluss weiterhin gewährleistet, wenn Zeile 6 und 7 in dem Algorithmus vertauscht werden? Erläutere den Grund hierfür. Modelliert diesen abgeänderten Algorithmus mit UPPAAL. Gebt gegebenenfalls eine Ausführungsreihenfolge an, die das Gegenteil beweist und fügt die zugehörige Trace-Datei der elektronischen Abgabe hinzu.

## Hinweise

Die Abgabe erfolgt als Ausdruck am Ende der Vorlesung und zusätzlich elektronisch über ein Subversion<sup>1</sup> Repository. Die Zugangsdaten zum Repository werden Euch per E-Mail zugeschickt. Die Dokumentation der Aufgabenlösung ist in LaTeX anzufertigen. Bitte vergesst nicht die Namen aller Gruppenmitglieder mitanzugeben. Die elektronische Abgabe soll alle mit UPPAAL erstellten Dateien, sowie die LaTeX Dateien der Dokumentation umfassen.

---

<sup>1</sup>Informationen zu Subversion findet ihr auf <http://subversion.tigris.org/>