

Übungen zur Vorlesung
Modern Concurrency Theory
Blatt 3

Prof. Dr. Roland Meyer
Anton Opaterny

Abgabe bis 22.05.2023 um 11:59 Uhr

Aufgabe 3.1 (Ableitung)

Betrachten Sie folgendes Programm:

```
1: [y := 3]
2: [z := 1]
3: if (even(z)) then
4:   [y := y + 1]
5: else
6:   [z := z + 1]
7: while (y > z) do
8:   [z := z + 1]
```

- Geben Sie die *Sig*-Struktur mit einer passenden Interpretation für das Prädikat *even* an. Das Prädikat soll 1 ausgeben, falls das Argument gerade ist und 0 sonst.
- Geben Sie die Small-Step Ableitung an. Die initiale Konfiguration ist $init = (c, (0, 0))$, wobei c das komplette Programm ist.

Aufgabe 3.2 (Divergenz)

Konstruieren Sie drei Programme, sodass je ein Programm eine der folgenden Bedingungen erfüllt.

- ein $\text{assume}(b)$ erreichbar ist, sodass b nicht erfüllt ist
- unendlich viele Konfigurationen erreichbar sind
- es einen erreichbaren Kreis im Konfigurationsgraphen gibt

Wie sieht die Small-Step Semantik der Programme aus (Beschreiben Sie nur Besonderheiten)?

Zeigen Sie, dass ein Programm eine divergierende / nicht-terminierende Berechnung hat, genau dann wenn es eine dieser drei Eigenschaften erfüllt.

Aufgabe 3.3 (Determinismus)

Zeigen Sie: Für alle $(c, \sigma) \rightarrow \sigma_1$ und $(c, \sigma) \rightarrow \sigma_2$ gilt $\sigma_1 = \sigma_2$.

**Abgabe in der Übung oder bis zum bis 22.05.2023 um 11:59 Uhr per Mail
an anton.opaterny@tu-braunschweig.de.**