

Übungen zu „Parallelität in funktionalen Programmiersprachen“, Sommer 2009

Nr. 8, Abgabe der Aufgaben: 16. Juni 2009 vor der Vorlesung

Aufgaben

8.1 Abstrakte Semantik

8 Punkte

Drücken Sie die Funktionen

(a) **filter**

```
filter :: (a → Bool) → [a] → [a]
filter p xs = [ x | x ← xs, p x ]
```

und

(b) **rev**¹

```
rev :: [a] → [a] → [a]
rev [] a = a
rev (x:xs) a = rev xs (x:a)
```

in Mini-Haskell aus.

Bestimmen Sie jeweils formal die abstrakte Semantik sowie die sicheren Auswertungsgrade der Argumente zu den möglichen Auswertungsgraden der Ergebnisse (*Evaluation Transformer* - $ET(\mathbf{filter})$ und $ET(\mathbf{rev})$).

8.2 Abstrakte Interpretation

4 Punkte

Schreiben Sie für jede Definiertheitsstufe $i \in \underline{4}$ eine Haskell-Funktion `evaluate_i` vom Typ $[a] \rightarrow ()$, die ihr Argument (eine Liste) zum gegebenen Grad i auswertet.

¹Mit Hilfe von `rev` ist **reverse** in GHC implementiert: `reverse xs = rev xs []`