

Deskriptive Prog, Aufgabe 32, 33

Christoph Schwerdtfeger

June 8, 2015

1 Aufgabe 32

1.1 LParserCore

$$\begin{aligned} \mathbb{Z} : (p \parallel q) ++ > f &= (p ++ > f) \parallel (q ++ > f) \\ &\quad (p \parallel q) ++ > fa \\ \iff \text{concatMap}(\lambda (x, y) \rightarrow f \ x \ y) \ (p \parallel q \ a) &\quad (\text{Def. } ++ >) \\ \iff \text{concatMap}(\lambda (x, y) \rightarrow f \ x \ y) \ (p \ a \ ++ \ q \ a) &\quad (\text{Def. } \parallel) \\ \iff \text{concat}(\text{map}(\lambda (x, y) \rightarrow f \ x \ y) \ (p \ a)) \ ++ \ \text{map}(\lambda (x, y) \rightarrow f \ x \ y) \ (q \ a)) &\quad (\text{Eigenschaft von map}) \\ \iff \text{concat}(\text{map}(\lambda (x, y) \rightarrow f \ x \ y) \ (p \ a)) \ ++ \ \text{concat}(\text{map}(\lambda (x, y) \rightarrow f \ x \ y) \ (q \ a)) &\quad (\text{Eigenschaft von concat}) \\ \iff (p ++ > fa) \parallel (q ++ > fa) &\quad (\text{Alles rückwärts}) \end{aligned}$$

1.2 ParserCore

$$\mathbb{Z} : (p \parallel q) ++ > f \neq (p ++ > f) \parallel (q ++ > f)$$

Mittels Gegenbeispiel - Betrachte folgende Definition:

```
p= many digit
q= many lower
f x y= if x == [()] then upper y else failure y
a = 'xY'
```

Dann wird $(p \parallel q) ++ > f$ im Gegensatz zur rechten Seite einen Fehler werfen (liegt an der Nutzung von yield in many).

2 Aufgabe 33

```
[x*x|x<-[1..150], x`mod`4 == 0]
[x * x | x `mod` 4 == 0,c]) [1..150])
```

```
concat (map (\x -> [x * x | x 'mod' 4 == 0]) [1..150])  
concat (map (\x -> if (x 'mod' 4 == 0) then (x * x) else []) [1..150])
```