
Concurrent Programming<http://www-pu.informatik.uni-tuebingen.de/cp-2006>

Blatt 4

Abgabe: 30.5.2006

1. [10 Punkte] Implementiere die in der Vorlesung vorgestellten synchronen Kommunikationskanäle. Ein Kanal soll als SRFI-9-Record repräsentiert werden. Programmiere die Operationen `send-sync` und `receive-sync` auf diesen Kanälen, um Daten in einen Kanal zu senden, bzw. Daten aus einem Kanal zu lesen.

Teste deine Implementierung mit den Beispielen aus der Vorlesung.

2. [5 Punkte] Erweitere die Lösung für das Producer/Consumer-Problem mit asynchronen Kanälen, so dass der Puffer eine beliebige Länge haben kann.
3. [15 Punkte] Schreibe eine Prozedur, die einen synchronen Kanal zurückgibt, auf dem aufsteigend sortiert alle Primzahlpotenzen ankommen.

Gehe dabei nach dieser Anleitung vor:

- (a) [1 Punkt] Programmiere zuerst eine Hilfsprozedur `receive-items`, welche die ersten n Nachrichten eines synchronen Kanals in einer Liste zurückgibt.

- (b) [2 Punkte] Schreibe eine Prozedur `powers`, die für eine Zahl n einen Kanal liefert, der die Potenzen von n enthält.

```
> (receive-items 10 (powers 2))  
(2 4 8 16 32 64 128 256 512 1024)
```

- (c) [2 Punkte] Schreibe eine Prozedur `channel-map` analog zu `map`:

```
> (receive-items 10 (channel-map (lambda (x) (+ x 1)) (powers 2)))  
(3 5 9 17 33 65 129 257 513 1025)
```

- (d) [4 Punkte] Schreibe eine Prozedur `channel-merge`, welche zwei Kanäle, die aufsteigende Folgen von Zahlen liefern, zu einem Kanal vereinigt. Die Folge von Zahlen im resultierenden Kanal soll ebenfalls aufsteigend sortiert sein.

```
> (receive-items 10 (channel-merge (powers 2) (powers 3)))  
(2 3 4 8 9 16 27 32 64 81)
```

- (e) [6 Punkte] Schreibe eine Prozedur `prim-power-channel`, die einen Kanal zurückgibt, der die aufsteigend sortierten Primzahlpotenzen liefert.