

Vertiefung Funktionale Programmierung: Clojure

Programmierübung 3: Transducers

Philipp Körner

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf

06. Mai 2024

Ihr solltet in der Lage sein,

- die Ideen hinter Transducern zu beschreiben.
- bestehende Transducer zu verstehen und korrekt zu verwenden.
- zustandslose Transducer zu schreiben.

Es sollen folgende Transformationen auf einer Liste von Zahlen durchgeführt werden:

- alle ungeraden Zahlen entfernen
- übrige Zahlen verdoppeln
- zwischen diese Zahlen ein Komma (Character `\,`) einfügen
- mit `str` zusammenführen

Was tut der folgende, statebehaftete Transducer? Was ist ein möglicher Aufruf?

```
(defn xyz []
  (fn [xf]
    (let [p (atom ::none)]
      (fn
        ([] (xf))
        ([a] (xf a))
        ([a e]
         (let [pp @p]
           (reset! p e)
           (if (= pp e)
              a
              (xf a e))))))))))
```

Transducer schreiben I

Implementieren Sie eine Funktion (`transplace m`), die eine Map `m` als Argument bekommt und einen Transducer zurückgibt. Wenn ein Element als Schlüssel in `m` vorhanden ist, soll es durch den assoziierten Wert ersetzt werden, ansonsten soll das originale Element verwendet werden. Die Funktion `replace` darf dabei nicht benutzt werden.

Beispielaufrufe:

```
user=> (transduce (transplace {:y :a}) conj [:x :y :z])
[:x :a :z]
user=> (transduce (comp (transplace {nil 0}) (map inc))
              conj
              [42 nil 3])
43 1 4]
user=> (transduce (comp (transplace {nil -1})
                      (partition-by pos?))
              conj
              [1 2 3 0 5 6])
[[1 2 3] [0] [5 6]]
```

Implementieren Sie einen Transducer (`transpeat n`), der alle Elemente `n` mal wiederholt.

Beispielaufruf:

```
user=> (transduce (transpeat 3) conj [:x :y :z])
[:x :x :x :y :y :y :z :z :z]
;; early termination
user=> (transduce (comp (take 3) (transpeat 2))
           conj [:x :y :z])
[:x :x :y :y :z :z]
user=> (transduce (comp (transpeat 2) (take 3))
           conj [:x :y :z])
[:x :x :y]
```