Heinrich-Heine-University Düsseldorf
Computer Science Department
Software Engineering and Programming Languages
Philipp Körner
Björn Ebbinghaus

Functional Programming – ST 2024
Reading Guide 03: Transducer

Timeline: This unit should be completed by 06.05.2024.

1 Material

- Material/repl2022-vertiefung/src/repl/16_transducer.clj
- Rich Hickey: Transducers `https://www.youtube.com/watch?v=6mTbuzafcII`
- Rich Hickey: Inside Transducers `https://www.youtube.com/watch?v=4KqUvG8HPYo`

2 Learning Outcomes

After completing this unit you should be able to

- describe the idea behind transducers.
- understand and correctly use existing transducers.
- write basic (non-stateful) transducers yourself.

3 Highlights

- reduce vs. transduce
- pass-through of step-functions
- state-flushing

4 Exercises

Exercise 3.1 (Transducer)
Implement a function `(transplace m)`, which receives a map `m` as argument and returns a transducer. If an element is present as a key in `m`, it is replaced by the associated value, otherwise the original element is retained.

The function `replace` must not be used for this.

Example calls:

```
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]]
```

Questions

If you have any questions, please contact Philipp Körner (`p.koerner@hhu.de`).