Higher Order Functions

Ken Q Pu, Faculty of Science, Ontario Tech University

Comprehensive Guide

https://clojure.org/api/cheatsheet

Ken Q Pu, Faculty of Science, Ontario Tech University

Higher-order functions

Higher-order functions are functions whose input and/or output are also functions.

Some examples of higher order functions:

- update
- apply

Update is a HOF

```
(update <hashmap> <key> <fn>)
```

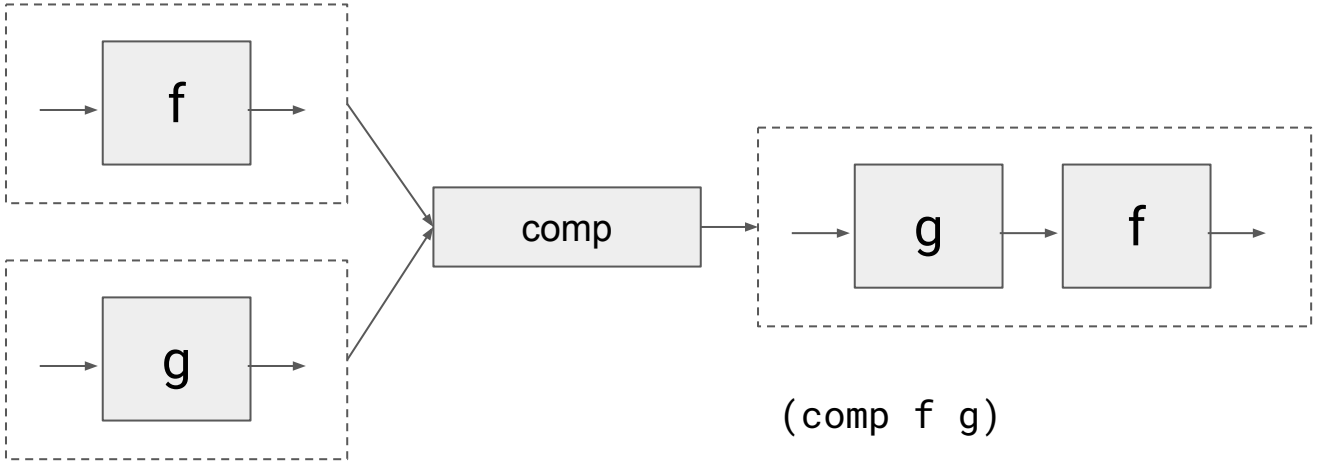
```
(apply <fn> <arguments>)
```

More higher order functions

These HOF return functions as output.

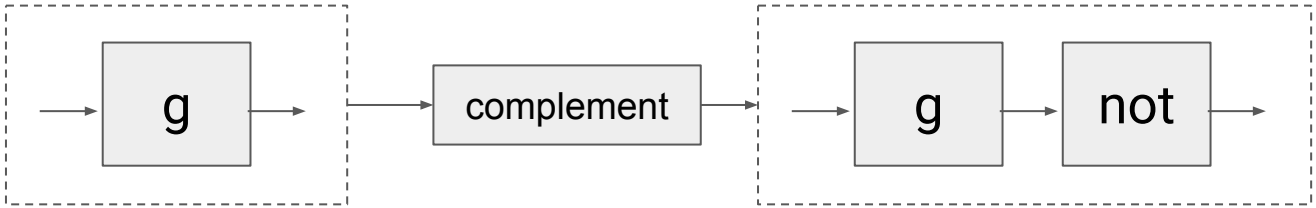
- comp
- complement
- partial
- fnil
- every-pred
- some-fn

comp: composing two functions




```
((comp (fn [x] (* 2 x)) inc) 5)
⇒ ((fn [x] (* 2 x)) (inc 5))
⇒ 2 * (5 + 1) = 12
```

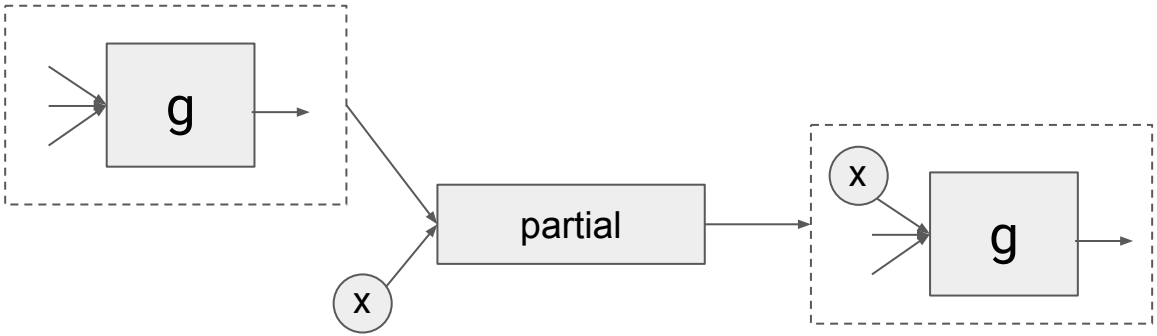
complement




```
(complement g)

(comp not g)
```

partial



partial

```
(def include-tax-price
 (fn [tax-rate price]
  (+ price (* tax-rate price))))


(def ontario-price (partial after-tax-income 0.13))

(def alberta-price (partial after-tax-income 0.05))
```

More HOF

Check out the documentation for

- fnil: protecting a function with unexpected nil arguments
- every-pred: generate logical conjunctive predicates
- some-fn: select a function that returns non nil