

Documentatie laborator 3 – Functii ca valori de ordinul 1. Functii curry/uncurry. Functionale.

Functii ca valori de ordinul 1

In programarea functionala functiile sunt valori de ordinul 1, ceea ce inseamna ca, fara vreun plus de efort, ele pot fi manipulate ca orice altfel de valoare. Mai exact, functiile pot fi:

- legate la un identificator (ex: (define par? even?))
- stocate intr-o structura de date (ex: '< > odd? even?))
- pasate ca argumente intr-un apel de functie (ex: (list? even?))
- returnate ca valoare a unui apel de functie (ex: functiile curry din paragraful urmator)

Functii curry/uncurry

Functie curry: functie care returneaza o noua functie atunci cand este aplicata pe mai putine argumente decat asteapta ea

Aplicatie partiala: aplicarea unei functii pe mai putine argumente decat asteapta ea; rezultatul este o noua functie (care asteapta restul de argumente)

Functie uncurry: functie care nu poate fi aplicata partial, trebuie sa isi primeasca toate argumentele deodata

Pentru exemple, consultati exercitiile 1-2 din fisierul lab3-doc.rkt. Functiile curry faciliteaza reutilizarea de cod, facilitand derivarea de functii particulare din functii mai generale (asa cum in exercitiul 1 am derivat functia increment din functia care aduna 2 numere).

Functionale

Cititi documentatia de la <http://mitpress.mit.edu/sicp/full-text/book/book-Z-H-12.html> pana la capitolul „[Using let to create local variables](#)”. Apoi parcurgeti rezumatul urmator.

Functionalele (higher order procedures – in engleza) reprezinta functii care manipuleaza alte functii (primindu-le ca argumente sau returnandu-le ca rezultat). Adesea ceea ce fac ele este sa surprinda patternuri de calcul. De exemplu:

- functionala map: (map f L) = lista rezultatelor aplicarii functiei f pe fiecare element din lista L (ex: (map odd? '(1 2 3 4 5)) intoarce (#t #f #t #f #t))
- functionala filter: (filter p L) = lista acelor elemente din L care satisfac predicatul p (ex: (filter odd? '(1 2 3 4 5)) intoarce (1 3 5))

Map sau filter sunt functionale care descriu patternul de a prelucra o serie de elemente intr-un acelasi fel, respectiv patternul de a filtra doar elementele care satisfac o anumita conditie. Acestea sunt prelucrari des intalnite, cu diverse functii (predicate) pe diverse liste

de elemente. Fara functionale ar trebui sa scriem mereu un cod asemanator, ceea ce nu este in spiritul reutilizarii codului. Ceea ce trebuie sa facem cand in spatele unor prelucrari observam un mecanism mai general, mai abstract, este sa scriem o functionala care descrie acel mecanism. Mai departe, toate cazurile particulare vor putea fi tratate mai repede si mai clar (va fi mai usor de inteles ce se intampla acolo) cu ajutorul acestei functionale.

Exercitiile 3-4 reprezinta implementarile functionalelor map si filter. Inainte de a le citi, incercati sa le realizati singuri.