

Operacje na listach c.d.

Materiały do ćwiczeń będą oparte w dużej mierze na materiałach dr. Sławomira Bakalarskiego (do tego samego kursu).

Funkcja `map` pozwala nam na wywołanie operacji na każdym elemencie listy. Co jednak, jeśli z listy argumentów chcemy otrzymać jeden wynik? Znamy już funkcje działające w ten sposób, na przykład `sum` czy `product`; o funkcjach `foldl` i `foldr` można myśleć jako o ich uogólnieniu.

Funkcja `foldl` wywołana na dwuargumentowym działaniu \circ , początkowej wartości x oraz liście elementów (x_1, \dots, x_n) zwróci nam wartość

$$(\dots ((x \circ x_1) \circ x_2) \circ \dots) \circ x_n.$$

Funkcja `foldr` działa analogicznie, jednak od prawej; wywołana na takich samych argumentach zwróci nam wartość

$$x_1 \circ (x_2 \circ \dots \circ (x_{n-1} \circ (x_n \circ x)) \dots).$$

foldl/foldr

W praktyce początkowa wartość to często element neutralny działania \circ . Na przykład, sumę listy możemy zaimplementować jako:

```
sumaListyLewa :: Num a => [a] -> a  
sumaListyLewa xs = foldl (+) 0 xs
```

Ponieważ dodawanie liczb rzeczywistych jest łączne i przemienne, definicja sumy jako

```
sumaListyPrawa :: Num a => [a] -> a  
sumaListyPrawa xs = foldr (+) 0 xs
```

zwróci nam (w skończonych przypadkach) te same wyniki.

foldl/foldr

Funkcje `foldl` i `foldr` są bardzo uniwersalne. Podobnie, jak w przypadku `map`, używając ewentualnie zdefiniowanych przez nas funkcji jesteśmy w stanie definiować złożone zachowania używając funkcji dwuargumentowej i listy. Jeśli chcielibyśmy na przykład sumować punkty na płaszczyźnie zamiast liczb, możemy zdefiniować funkcję

- o `:: Num a => (a,a) -> (a,a) -> (a,a)`
- o `(a,b) (c,d) = (a+c,b+d)`

Funkcją sumującą listę par po współrzędnych będzie wtedy `foldl o (0,0)` (oczywiście `foldr` da nam tutaj ten sam wynik). Działa to oczywiście dużo ogólniej, jeśli np. mielibyśmy zdefiniowany własny typ reprezentujący macierze oraz dwuargumentową operację mnożenia macierzy, jesteśmy w stanie w jednej linijce zdefiniować mnożenie całej listy macierzy.