



Kurs języka Haskell 2024/25

LISTA NR 9 (TERMIN: 20.01.2025, godz 5:00)

Uwaga: Wszystkie rozwiązania należy umieścić w jednym module o nazwie `Lista9` zachowując sygnatury zgodne z szablonem rozwiązań zamieszczonym w SKOS-ie.

Zadanie 1. Jedną z wielu wersji zbalansowanych drzew BST są drzewa balansowane na podstawie rozmiaru poddrzew. Jeśli jedno z poddrzew jest większe o ustalony czynnik, następuje rotacja. Takie drzewo można zdefiniować jako następujący typ danych:

```
type Size = Int

data Tree a = Leaf | Node Size a (Tree a) (Tree a)

size :: Tree a → Size
size Leaf = 0
size (Node s _ _ ) = s
```

Przykładową rotację można zaimplementować tak:

```
singleL :: Tree a → Tree a
singleL (Node _ a l (Node _ b rl rr)) =
  let newl = Node (size l + size rl + 1) a l rl
  in Node (size newl + size rr + 1) b newl rr
```

Implementację wszystkich potrzebnych rotacji wraz z wstawianiem elementu można znaleźć w szablonie rozwiązań. Zmodyfikuj kod z szablonu tak, by używał on sprytnego konstruktora (zarówno do konstruowania drzew, jak i w dopasowaniu wzorca) tak, by użytkownik nie musiał już nigdy samodzielnie ustawiać wielkości poddrzew w konstruktorze. Użyj rozszerzenia `PatternSynonyms`.

Zadanie 2. Przeczytaj artykuł Stephena Adamsa o drzewach balansowanych przez rozmiar poddrzew¹. Zaimplementuj operację scalania dwóch drzew (implementacja w SML-u jest pokazana w artykule). Dodaj sprytny konstruktor używając rozszerzenia `PatternSynonyms`, który pozwala użytkownikowi postrzegać takie drzewo jak zwykłe drzewo BST, podczas gdy balansowanie odbywa się w całości wewnątrz sprytnego konstruktora.

Zadanie 3. Zmodyfikuj kod z wykładu 10. o typach egzystencjalnych (kształty) tak, by móc zaimplementować funkcję

```
sameShape :: (IsShape a, IsShape b) ⇒ a → b → Bool
```

która porównuje czy dwie wartości w klasie `IsShape` są tym samym kształtem. Nie porównuj wartości zwracanych przez metodę `name`, a zamiast tego użyj modułu `Data.Typeable`, który udostępnia klasę `Typeable`. Dla wartości typów znajdujących się w tej klasie, można użyć funkcji `typeOf`, która zwraca runtime'ową reprezentację typu wartości. Nie jest dla nas istotne, jak ta reprezentacja wygląda, ważne, że jest ona w klasie `Eq`. Dodatkowo, kompilator sam umie zainstalować typ w tej klasie przy użyciu słowa kluczowego `deriving`. Przykładowo:

```
ghci> let s1 = shapeFactory2 SCircle 2
ghci> let s2 = shapeFactory2 SCircle 3
ghci> let s3 = shapeFactory2 SRect (2, 3)
ghci> sameShape s1 s2
True
ghci> sameShape s1 s3
False
```

Wskazówka: Dodaj do klasy `IsShape` metodę zwracającą reprezentację typu. Jaka powinna być jej implementacja dla typu `Shape`?

Zadanie 4. Zaimplementuj *tabulation hashing*² dla typu `Word64`. Implementacja powinna składać się z funkcji

```
hashWithTable :: Int → StdGen → Word64 → Word64
```

gdzie:

- `hashWithTable n seed` tworzy losową tablicę przy użyciu ziarna `seed` dla haszowania do `n` bitów i zwraca funkcję haszującą przy użyciu tej tablicy.
- Użyj tablic z modułu `Data.Array`.
- `StdGen` znajduje się w module `System.Random`.
- `Word64` znajduje się w module `Data.Word`.
- Operacje na bitach zdefiniowane są w module `Data.Bits`.

Zadanie 5. Użyj haszowania do 6 bitów z poprzedniego zadania do zaimplementowania w monadzie `ST` filtru Bloom³ dla zbioru wartości typu `Word64`, w którym tablica bitów ma 64 elementy (może więc być reprezentowana przez

¹https://www.cambridge.org/core/services/aop-cambridge-core/content/view/0CAA1C189B4F7C15CE9B8C02D0D4B54E/S0956796800000885a.pdf/functional_pearls_efficient_setsa_balancing_act.pdf

²https://en.wikipedia.org/wiki/Tabulation_hashing#Method

³https://en.wikipedia.org/wiki/Bloom_filter

pojedyncze słowo maszynowe). By uzyskać n funkcji haszujących, użyj `hashWithTable` dla ziaren utworzonych statycznie przy użyciu funkcji `mkStdGen` z modułu `System.Random`. Niech liczba funkcji haszujących będzie parametrem tworzenia filtra.

Zaimplementuj:

```
data Bloom s
createBloom :: Int → ST s (Bloom s)
insertBloom :: Word64 → Bloom s → ST s ()
elemBloom  :: Word64 → Bloom s → ST s Bool
```