

## Zadanie zaliczeniowe – Why3 Element wspólny tablic i wpisywanie drzewa do listy

Termin rozwiązania zadania: 30 maja 2014

### 1 Wstęp

To zadanie zaliczeniowe składa się z trzech zadań do wykonania w Why3: `wspolny`, `toList` i `bstToList`. Zadania będą polegały na napisaniu odpowiednich funkcji i specyfikacji tych funkcji tak aby wszystkie obligacje dowodowe wygenerowane przez Why3 można było udowodnić automatycznie.

### 2 Wspólny

W pliku `wspolny.mlw` znajduje się nagłówek funkcji `wspolny (a: array int)(b: array int) : bool`. Zadanie polega na napisaniu treści i specyfikacji tej funkcji. Funkcja ma sprawdzać czy dwie posortowane niemalejąco tablice `a` i `b` mają element wspólny. Tablice `a` i `b` mogą mieć różne długości.

### 3 ToList

#### 3.1 ToList

To zadanie jest tylko częścią zadania `bstToList`. Jest wydzielone w osobnym pliku, żeby łatwiej z nim eksperymentować.

W pliku `toList.mlw` znajdują się definicje:

- typu `tree` drzew binarnych o wartościach całkowitych w węzłach,
- predykatu `contains x t` stwierdzającego, że w drzewie `t` występuje wartość `x`,
- funkcji `size t` obliczającej rozmiar drzewa `t`.
- funkcji `toListA` i `toListA`.

Dodatkowo są tam dwa lematy `contains_node` i `size_nonneg`. W sesji znajdującej się w katalogu `toList` oba te lematy są już udowodnione i nie trzeba ich dowodzić.

Zadanie polega na napisaniu specyfikacji funkcji:

1. `toList t` przepisującej elementy drzewa `t` do listy, poprzez wywołanie `toListA t Nil`,
2. `toListA t a` przepisującej elementy drzewa `t` do listy, używając akumulatora `a`.

#### 3.2 BstToList

Plik `bstToList.mlw` jest rozszerzeniem pliku `toList.mlw`. Zawiera dodatkowo:

- definicję predykatu indukcyjnego `bst t` opisującego kiedy drzewo `t` jest BST,

- lematu `bst_Node_inv`. Podobnie jak poprzednio wspomniane lematy jest on już udowodniony w sesji znajdującej się w odpowiednim katalogu (`bstToList`).

Zadanie polega na napisaniu specyfikacji funkcji:

1. `toList t` przepisującej elementy drzewa BST `t` do listy, poprzez wywołanie `toListA t Nil`. Specyfikacja powinna zapewniać, że wynikowa lista jest posortowana. Proszę zwrócić uwagę na kolejność elementów w drzewie (malejąca) i w wynikowej liście (rosnąca).
2. `toListA t a` przepisującej elementy drzewa BST `t` do listy, używając akumulatora `a`.

## 4 Rozwiązania

Rozwiązaniem zadania jest odpowiedni plik `.mlw` wraz z katalogiem zawierającym sesję `why3ide`. Rozwiązania proszę przesyłać do mnie emailem.