

Lista zadań nr 13

Zadania dotyczące dowodów indukcyjnych można prezentować na tablicy (dowody tradycyjne), lub przy użyciu asystenta dowodzenia Coq/Rocq.

Zadanie 1. (2 pkt)

Rozważmy algebraiczne typy danych, które nie mają parametrów, a typy argumentów konstruktorów nie mogą być typami złożonymi. W OCamlu definicje takich typów danych można opisać następująco.

```
type ty_name = string
type ctor_name = string

type ctor = Ctor of ctor_name * ty_name list
type ty_def = ty_name * ctor list
```

Na przykład typ list wartości typu int opiszemy następująco.

```
("int_list",
 [ Ctor ("Nil", []);
   Ctor ("Cons", ["int"; "int_list"])
 ])
```

Napisz funkcję, która dla podanej definicji typu drukuje zasadę indukcji dla tego typu.

Zadanie 2. (2 pkt)

Zdefiniuj predykat (relację unarną) mówiący, że lista jest posortowana niemalejąco.

Zadanie 3. (2 pkt)

Zaimplementuj sortowanie przez wstawianie. Następnie pokaż, że wynikowa lista jest posortowana niemalejąco.

Zadanie 4. (2 pkt)

Przyjmij następującą definicję relacji mówiącej, że lista jest permutacją innej listy.

$$\begin{array}{c}
 \frac{}{[] \sim []} \qquad \frac{xs \sim ys}{x :: xs \sim x :: ys} \\
 \frac{xs \sim ys \quad ys \sim zs}{xs \sim zs} \qquad \frac{}{x :: y :: xs \sim y :: x :: xs}
 \end{array}$$

Pokaż, że jest to relacja równoważności.

Zadanie 5. (2 pkt)

Pokaż, że funkcja z zadania 3 zwraca permutację listy wejściowej. Użyj definicji permutacji z poprzedniego zadania.

Zadanie 6. (1 pkt)

Sformułuj zasadę indukcji dla relacji z zadania 4.

Zadanie 7. (1 pkt)

Zaproponuj formalną specyfikację funkcji `to_nnf` z poprzedniej listy zadań.