

# Programowanie Funkcyjne 2018

Lista zadań nr 6

21 listopada 2018

Zadania na ten tydzień dotyczą sprawdzania poprawności dowodów w systemie dedukcji naturalnej dla rachunku zdań, który został wprowadzony na poprzedniej liście zadań. W systemie SKOS znajdziecie szkielet rozwiązania, zawierający następujące pliki:

- `Syntax.ml` zawierający składnię formuł i dowodów (w dwóch wersjach, jako drzewo i skrypt wprowadzony na poprzedniej liście zadań)
- `Lexer.mll`, `Lexer.mli` i `Parser.mly`, które wspólnie definiują lekser i parser, wczytujący dowody z pliku tekstowego do postaci zdefiniowanej w module `Syntax`
- `PC.ml` który jest głównym modulem systemu sprawdzania dowodów
- `example.pf` zawierający przykładowe dowody z poprzedniej listy, zapisane jako drzewo i jako skrypt.

Cały projekt najłatwiej skompilować przy użyciu `ocamlbuild`, wydając polecenie `ocamlbuild PC.native`. Program wczytuje dane ze standardowego wejścia, więc możemy go uruchomić na przykład tak:  
`./PC.native < example.pf`.

**Zadanie 1 (6 pkt).** Dostarczony projekt to, oczywiście, tylko szkielet który wczytuje z pliku dowody i wypisuje na standardowe wyjście ich liczbę. Zaimplementuj algorytm sprawdzający poprawność dowodów *w postaci drzewiastej* jako funkcję przyjmującą jako argument wartość typu `string` `pt`, i rozbuduj moduł `PC` tak aby sprawdzał poprawność wszystkich wczytanych dowodów w postaci drzewiastej (dowody w postaci skryptu program może pomijać) i informujący użytkownika o poprawności każdego z dowodów lub o rodzaju znalezionej błęd.

**Zadanie 2 (10 pkt).** Zaimplementuj algorytm sprawdzający poprawność dowodów w postaci skryptu jako funkcję przyjmującą jako argument wartość typu `string` `ps` i rozbuduj moduł `PC` analogicznie do poprzedniego zadania. W przypadku rozwiązania obydwu zadań program powinien działać również dla danych wejściowych zawierających obydwa rodzaje dowodów.

**Zadanie 3 (4 pkt).** Przetestuj wyczerpująco swój program, tj. dostarcz plik (lub pliki) z ciekawymi dowodami (zarówno poprawnymi jak i niepoprawnymi) ilustrującymi jak działa Twoja implementacja.

**Uwaga:** nazwy celów (twierdzeń) składają się wyłącznie z małych liter i podkreśleń, zaś nazwy zmiennych – wyłącznie z wielkich liter. Nazwy poszczególnych węzłów drzewa dowodu odpowiadają nazwom konstruktorów typu `'a` `pt`.