

Lista 4

Algebraiczne Typy Danych

W poniższych zadaniach dopuszczalne jest wykorzystanie funkcji wbudowanych obliczających długość listy, odwracających listę oraz łączących dwie listy, o ile nie wpływają one na drastyczne pogorszenie złożoności obliczeniowej.

Każde zadanie, poza implementacją funkcji, musi posiadać **kompletny zestaw testów**.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 4.

- 1) Jednym z bardzo istotnych, nowoczesnych narzędzi stosowanych podczas wytwarzania oprogramowania są systemy kontroli wersji (*ang. version control system*). Pozwalają one na przechowywanie całości projektu wraz z historią zmian (repozytorium), dzięki czemu programista może dowolnie testować pomysły, wprowadzać nowe zmiany, wycofywać te, które są nietrafione oraz przeglądać całą historię rozwoju projektu.

Większość systemów kontroli wersji opiera się o ideę migawek (*ang. snapshot*). W tym podejściu, na życzenie użytkownika, tworzona jest migawka projektu zawierająca zapis wszystkich zmian wprowadzonych od poprzedniej migawki. Dzięki temu, można łatwo zobaczyć, które fragmenty programu zmieniono oraz kto to zrobił, a sama reprezentacja pozwala znacząco ograniczać zużycie pamięci. Odtworzenie historycznej wersji projektu polega na przejściu po historii zmian do danej migawki i ich zaaplikowaniu.

Zadanie polega na stworzeniu prymitywnego systemu kontroli wersji dla łańcucha znaków. Wykonaj następujące polecenia: (OCaml i Scala)

- a. Zaproponuj i zaimplementuj algebraiczne typy danych (jeden lub więcej), pozwalające na: (20pkt.)
 - Reprezentację repozytorium jako drzewa kolejnych migawek,
 - Reprezentację operacji migawkowych: utworzenia łańcucha znaków, wstawienia podłańcucha, usunięcia podłańcucha, przeniesienia fragmentu łańcucha w inne miejsce (fragment z pozycji *i* zostaje wycięty i wstawiony na pozycję *j* łańcucha bez tego fragmentu).
- b. Napisz funkcję *review* zwracającą łańcuch będący efektem zastosowania przekazanej migawki (uwzględniając oczywiście migawki poprzednie). (30pkt.)

Ze względu na to, że implementacja funkcji *review* może wymagać utworzenia kilku funkcji pomocniczych, mogą one być zdefiniowane poza funkcją *review*. Jeżeli którakolwiek migawka zawiera **błąd** tzn. nie ma możliwości wykonania operacji na aktualnym łańcuchu, to należy zwrócić **pustą wartość typu opcjonalnego**.