

Programowanie funkcyjne 2014

Lista 6

kzi

25 listopada 2014

Uwaga: Jeśli zadanie jest niedospecyfikowane, to braki w specyfikacji można uzupełnić w dowolny sensowny sposób.

Zadania w OCaml'u

W zadaniach nie można korzystać z pętli, można za to (a raczej trzeba) korzystać z modyfikowalnych struktur danych.

1. Załóżmy, że w języku nie ma modyfikowalnych tablic. Zdefiniuj za pomocą referencji jakiś zamiennik — wystarczy typ — oraz odpowiednik tablicy `[|0;2;3|]`. Następnie podstaw w nim 1 zamiast 0. (2ptk)
2. Zdefiniuj typy `'a graph = 'a list * ('a -> 'a list)`,
`raw_graph = int * (int -> int list)`,
`'a opt_graph = raw_graph * (int -> 'a) * ('a ->int)`
dla grafów skierowanych w postaci list sąsiedztwa. (0ptk)
3. Zdefiniuj funkcję `raw_graph : 'a graph -> raw_graph`, która zaaplikowana do grafu $([v_1, \dots, v_n], e)$ zwraca graf (n, e') taki, że
 - dla każdego $i \in \{1, \dots, n\}$ zachodzi: j jest elementem $e' i$ wtw v_j jest elementem $e v_i$; oraz
 - funkcja e' działa w czasie stałym. (3ptk)
4. Zdefiniuj funkcję `opt_graph : 'a graph -> 'a opt_graph`, która zaaplikowana do grafu $G = ([v_1, \dots, v_n], e)$ zwraca graf $(\text{raw_graph } G, \text{lab}, \text{nod})$ taki, że
 - dla każdego $i \in \{1, \dots, n\}$ zachodzi: $\text{lab } i = v_i$ oraz $\text{nod } v_i = i$.Jedna z funkcji `lab`, `nod` powinna działać w czasie stałym, druga w nie gorszym niż liniowy od ilości wierzchołków. (2ptk)
5. Zdefiniuj funkcję `short_path : 'a opt_graph -> 'a -> 'a -> 'a list option`, która działa tak jak ta z listy 4, ale w czasie liniowym. (5ptk)
6. Napisz program, który bierze jako argumenty ścieżkę do pliku f i wyrażenie regularne r , i wypisuje wszystkie linie z f , które zawierają wzorzec r łącznie z ich numerami. Po wypisaniu każdej linii program powinien wczytać linię ze standardowego wejścia i jeśli jest to literka "c" (cancel) powinno się zamknąć. Linie powinny być wyświetlane w odwrotnej kolejności w stosunku do ich wystąpienia w f . (7ptk)
Wskazówka: Biblioteka do obsługi wyrażeń regularnych to Str. Funkcje do obsługi plików znajdują się w bibliotece Pervasives.
7. Wygeneruj losowy plik tekstowy z niezbyt długimi liniami i przetestuj program z zadania 4. Plik powinien być w miarę duży, tzn. kilkaset linii. (Możesz skorzystać z funkcji `gen_int_list` z listy 2.) (2ptk)