

Lista 7

Abstrakcyjne typy danych

W poniższych zadaniach **nie wolno** wykorzystywać funkcji wbudowanych obliczających **długość listy**, **odwracających listę** oraz **łączących dwie listy**.

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

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 7.

- 1) Koprocesor jest rodzajem procesora dedykowanego do wykonywania obliczeń zmiennoprzecinkowych. Często jest on realizowany w postaci tzw. maszyny stosowej. Maszyna stosowa jest procesorem, którego rejestry zorganizowane są w stos, a operacje wykonywane są jedynie na wierzchołku stosu i, w razie potrzeby, drugim jego elemencie (por. Fig. 1).

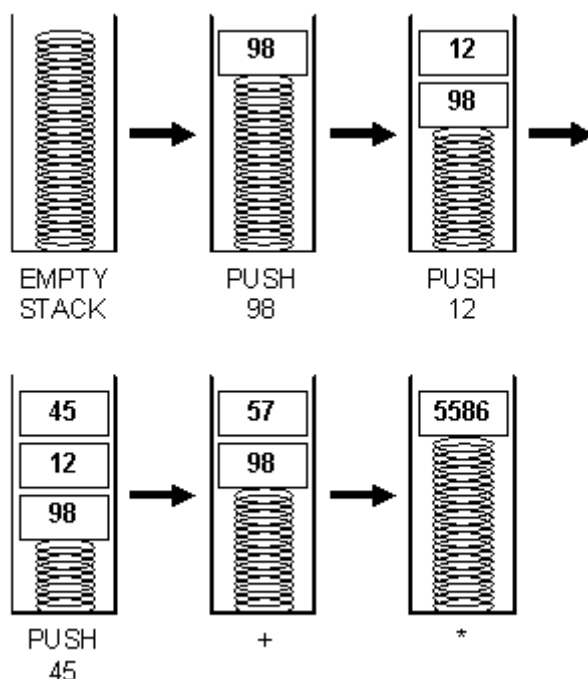


Fig. 1 Przykład działania maszyny stosowej,
źródło: https://users.ece.cmu.edu/~koopman/stack_computers/fig3_2.gif

Wykorzystując moduły języka OCaml zdefiniuj:

- a. Moduł *StackMachine* implementujący **modyfikowalną** maszynę stosową. Moduł ma definiować:
 - Typ wewnętrzny reprezentujący stos rejestrów, (1 pkt)
 - Wyjątki dla błędów: dzielenia przez zero oraz braku odpowiedniej liczby argumentów dla operacji na stosie, (1 pkt)
 - Funkcję *init* tworzącą nową maszynę stosową, (1 pkt)

- Funkcje pomocnicze pozwalające wykonywać instrukcje koprocatora zdefiniowane typem: (8 pkt.)

type instruction = Rst | LoadF of float | LoadI of int | Cpy | Add | Sub | Mul | Div;;

Gdzie Rst oznacza wyczyszczenie stosu, a Cpy wstawienie na stos kopii wierzchołka.

- Funkcję *result* pozwalającą odczytać wierzchołek stosu, (1 pkt)
 - Funkcję *execute* wykonującą listę instrukcji. (2 pkt.)
- b. Interfejs *COPROCESSOR* ukrywający szczegóły implementacyjne koprocatora. Koprocator powinien umożliwiać inicjalizację, a także wykonanie listy rozkazów i odczytanie wyniku, (4 pkt.)
- c. Moduł Coprocessor powstały z modułu StackMachine w wyniku ukrycia szczegółów implementacyjnych. (2 pkt.)