

Отчет по лабораторной работе № 4 по курсу «Функциональное программирование»

Студент группы М8О-307-19 МАИ *Тимофеев Алексей Владимирович*, №21 по списку

Контакты: `TimofeevAV8f@yandex.ru`

Работа выполнена: 13.05.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Знаки и строки.

2. Цель работы

Цель работы: научиться работать с литерами (знаками) и строками при помощи функций обработки строк и общих функций работы с последовательностями.

3. Задание (Вариант 2.39)

Вариант 4.40 (сложность 3) Запрограммировать на языке Коммон Лисп функцию, принимающую один аргумент - дерево, т.е. список с подписками, представляющий форму арифметического выражения Лисп. В выражении допустимы только четыре арифметические функции $+$, $-$, $*$ и $/$ (предусмотреть случай унарных функций $-$ и $/$).

символы переменных,

числовые константы.

Функция должна вернуть строку этого арифметического выражения в постфиксной польской записи в предположении, что все арифметические операторы в ней трактуются как бинарные.

$(\text{form-to-postfix } '(+ (* b b) (- (* 4 a c)))) \Rightarrow "b b * 0 4 a c * * - +"$

4. Оборудование студента

Процессор Intel Core i5-10600K @ 4.10GHz, память: 16 Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 20.04.4 LTS, компилятор GNU CLISP 2.49.92, текстовый редактор VS Code

6. Идея, метод, алгоритм

Функция получает дерево арифметических выражений Лиспа в виде списка списков. Данный список списков проверяется на несколько случаев:

1. Если в подсписке два элемента и первый из них `-`, значит это унарная операция разности.
2. Если в подсписке два элемента и первый из них `/`, значит это унарная операция деления.
3. Если нет совпадения с двумя верхними случаями, то сразу передаем текущий элемент списка и следующий в функцию `f`.

В функции `f` с помощью функции `g` проверяем, является ли элемент атомом, и далее печатаем в виде обратной польской нотации.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
; 4.40
(defun prior (op)
  (cond
    ((eq op '+) 1)
    ((eq op '-') 2)
    ((eq op '*') 3)
    ((eq op '/') 4)
    (t 0)
  )
)

(defun non-associative (op)
  (or (eq op '-') (eq op '/))
)

(declaim (ftype (function (t t) t) g))

(defun f (op l)
  (if (cdr l)
      (concatenate 'string (g (car l) op) " " (f op (cdr l)))
      " (write-to-string op))
)
```

```

        (g (car l) op)
    )
)

(defun h (l external-op)
  (cond
    ((= (list-length l) 2)
     (cond
       ((eq (car l) '-')
        (concatenate 'string "0 " (f (car l) (cdr l))
         " -"))
       ((eq (car l) '/')
        (concatenate 'string "1 " (f (car l) (cdr
l))) " /"))
       (t
        (f (car l) (cdr l)) )
      )
    )
    (t (f (car l) (cdr l))))
  )
)

(defun g (l external-op)
  (if (atom l)
      (write-to-string l)
      (h l external-op)
    )
  )
)

(defun form-to-postfix (l)
  (h l '+)
)

;; (form-to-postfix '(+ (* b b) (- (* 4 a c))))
;; (form-to-postfix '(* (* a b) (* c (* d e))))
;; (form-to-postfix '(/ (- b c d) a) )

;; (form-to-postfix '(- (+ A (- (* B C))) D) )
;; (form-to-postfix '(* (+ A B) (+ C D)) )
;; (form-to-postfix '(- (+ (+ A B) C) D) )
;; (form-to-postfix '(+ (* A B) (* C D)) )

```

```
;; (form-to-postfix '(* (/ a) c) )
```

8.2. Результаты работы

```
[1]> (load "lab4.lisp")
;; Loading file lab4.lisp ...
;; Loaded file lab4.lisp
#P"/mnt/d/education/education/FP/Mylab/lab4/lab4.lisp"
[2]> (form-to-postfix '(+ (* b b) (- (* 4 a c))))
"B B * 0 4 A C * * - +"
[3]> (form-to-postfix '(* (* a b) (* c (* d e))))
"A B * C D E * * *"
[4]> (form-to-postfix '(- (+ A (- (* B C))) D) )
"A 0 B C * - + D -"
[5]> (form-to-postfix '(* (/ a) c) )
"1 A / C *"
```

9. Дневник отладки

| Дата | Событие | Действие по исправлению | Примечание |
|------|---------|-------------------------|------------|
|------|---------|-------------------------|------------|

10. Замечания автора по существу работы

Довольно интересная задача. Эта задача мне показалась сложнее попавшихся мне ранее. Из-за обилия различных случаев «взаимодействия» операторов с их приоритетами и свойствами.

11. Выводы

При выполнении лабораторной работы № 4 я познакомился со списками языка Common Lisp, я научился работать с литералами и строками в языке Коммон Лисп. Написанная программа работает правильно и прошла все тесты.