Отчет по лабораторной работе № 5
по курсу «Функциональное программирование»

Студент группы 8о-306Б МАИ *Макаров Никита*, №11 по списку
Контакты: quizbeat@gmail.com
Работа выполнена: 22.05.2015

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

Знаки, строки и символы.

2. Цель работы

Научиться работать с литерами (знаками) и строками при помощи функций обработки строк и общих функций работы с последовательностями, использовать свойства символов.

3. Задание

Запрограммировать на языке Коммон Лисп функцию, принимающую один аргумент - дерево, т.е. список с подсписками, представляющий форму арифметического выражения Lisp. 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.

```
(form-to-postfix '(+ (* b b) (- (* 4 a c)))) => "b b * 4 a c * - +"
```

4. Оборудование студента

Процессор Intel Core i5 2 @ 1.3GHz, память: 4Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Mac OS X 10.9, среда Clozure CL 1.10

6. Идея, метод, алгоритм

Небольшое замечание: результат работы функции в примере задания не является корректным выражением в обратной польской записи, так как, во-первых, упущен унарный минус, а во-вторых, недостаточно операторов для обработки выражения.

Обратная польская нотация — форма записи математических выражений, в которой операнды расположены перед знаками операций. Также именуется как обратная польская запись, обратная бесскобочная запись, постфиксная нотация и т.п.

Так как требуется перевести арифметическое Lisp-выражение, то будем обрабатывать последовательности вида `(оператор список-операндов)`. В случае, когда длина списка операндов равна 1, оператор является унарным и он обрабатывается довольно просто. Для оператора `/` создается строка вида `''1 операнд /''`. Для унарного `-` добавляется `0`, чтобы выразить отрицание и получается строка вида `''0 операнд -''`. Для обработки нескольких операндов функция `proc-args` рекурсивно создает строку, состоящую из приведенного к постфиксной записи первого элемента списка операндов и результата вызова этой же функции от хвоста списка операндов, добавляя на каждом вызове текущий оператор. Все операнды приводятся к постфиксному виду. Функция `single-arg` проверяет, является ли текущий оператор унарным.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
(defun single-arg (expr) ;; checks for unary operator
  (and (not (null expr)) (not (null (rest expr))) (null (rest
   (rest expr))))
)

(defun form-to-postfix (expr)
  (if (listp expr)
      (if (single-arg expr)
          (if (eq (first expr) '- )
              (concatenate 'string "0 "(princ-to-string
   (form-to-postfix (second expr))) " -")
              (concatenate 'string "1 " (princ-to-string
   (form-to-postfix (second expr))) " /")
          )
          (concatenate 'string (form-to-postfix (second expr))
   " "
                               (proc-args (first expr) (rest
   (rest expr))))
      )
```

```
        (string-downcase (princ-to-string expr))
    )
)

(defun proc-args (operator expr) ;; processes the given
    arguments
    (if (not (null expr))
        (if (null (rest expr))
            (concatenate 'string (form-to-postfix (first expr)) "
    "
                                 (princ-to-string operator))
            (concatenate 'string (form-to-postfix (first expr)) "
    "
                                 (princ-to-string operator) " "
                                 (proc-args operator (rest expr)))
        )
    )
)

(defun t (expr) (form-to-postfix expr))
```

8.2. Результаты работы

```
> (form-to-postfix '(+ (*  b b) (- (* 4 a c))))
"b b * 0 4 a * c * - +"
> (form-to-postfix '(* (- 2) (/ 3)))
"0 2 - 1 3 / *"
> (form-to-postfix '(+ 7 (* 2 3)))
"7 2 3 * +"
> (form-to-postfix '(* (+ (- 2) (* 5 6)) (/ (- 9 3))))
"0 2 - 5 6 * + 1 9 3 - / *"
> (form-to-postfix '(+ (- (* (/ a) 2 c 4) e 5 g) 7 i))
"1 a / 2 * c * 4 * e - 5 - g - 7 + i +"
```

9. Дневник отладки

10. Замечания, выводы

В языке Common Lisp есть неплохие средства для работы со строками и символами, такие как конкатенация, удаление заданных символов, проверка совпадений, поиск и прочие. Применив рекурсию и некоторые из упомянутых выше функции можно легко реализовать программу, преобразовывающую арифметическое выражение в какую-либо нотацию. В данной лабораторной работе производится преобразование в обратную польскую нотацию, которая немного напоминает запись Lisp-выражений, но располагает операторы справа от операндов.