

Выпускная квалификационная
работа:

Оптимизация результатных
выражений в компиляторе
языка Рефал

Копьев Е.В.

Цель работы

Необходимо оптимизировать построение результатных выражений в компиляторе Простого Рефала.

Постановка задачи

Дано:

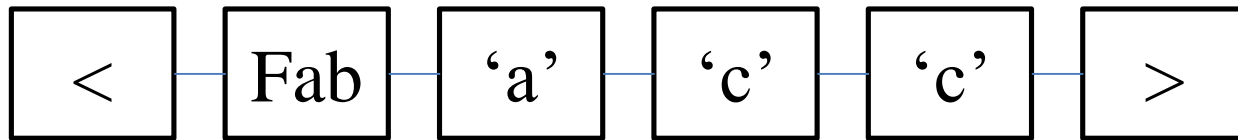
1. Входные данные: двусвязный список объектов (поле зрения).
2. Выходные данные: необходимый образ этого списка (результатное выражение).

Требуется:

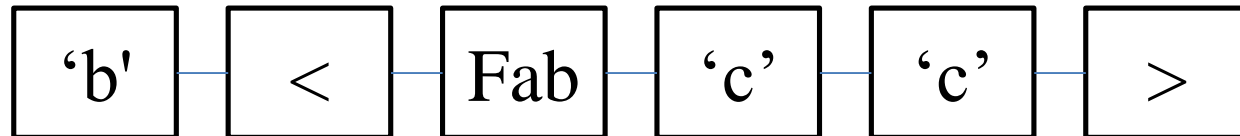
1. Задать преобразование, отображающее заданный список в заданный образ.
2. Минимизировать вычислительные затраты на это преобразование.

Пример исходных данных

Поле зрения (заданный список)



Результат (необходимый образ)



Элементарные преобразования в текущей реализации

1. Создание элемента в списке свободных узлов.
2. Перенос диапазона элементов.

Текущая реализация построения результатного выражения

Список свободных узлов:

Поле зрения:



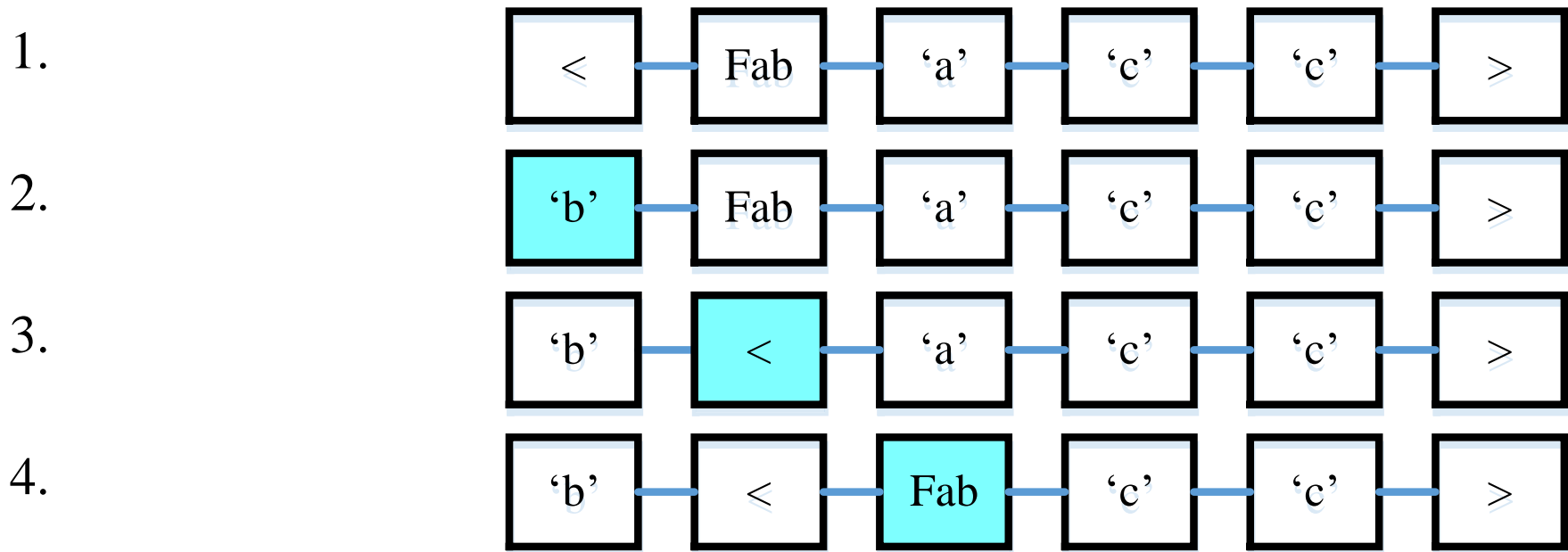
Элементарные преобразования в оптимизированной реализации

1. Создание элемента в списке свободных узлов.
2. Перенос диапазона элементов.
3. Переопределение содержимого узла.

Оптимизированная генерация результатного выражения

Список свободных узлов:

Поле зрения:



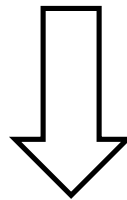
Алгоритм GST

```
tile_exists = True
Tiles = []
while tile_exists:
    tile, tile_exists =
        find_tile(Pat, Res)
    if tile_exists:
        mark_tile(Pat, Res, tile)
        Tiles += tile
return Tiles
```

Пример

Необходимо выполнить преобразование:

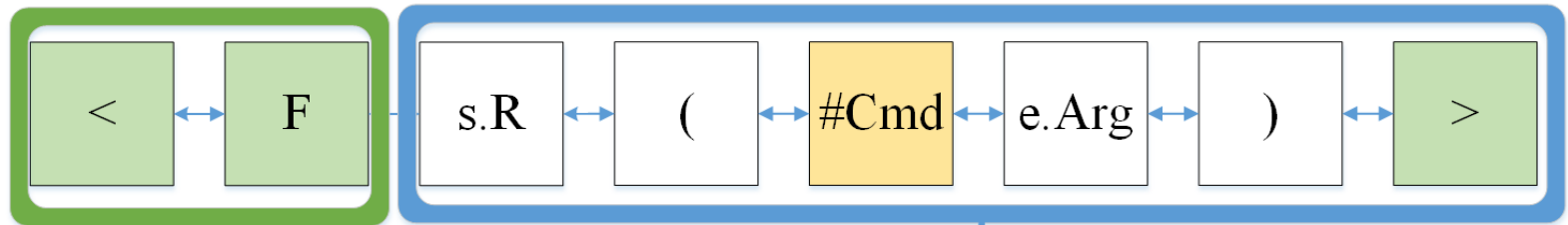
$\langle F \ s.R \ (\#Cmd \ e.Arg) \rangle$



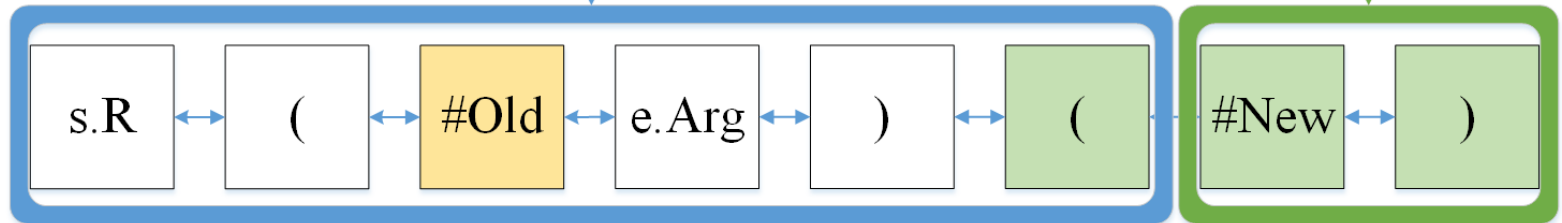
$s.R \ (\#Old \ e.Arg) \ (\#New)$

Пример найденного перекрытия

Вызов функции
(исходный список):



Результат вызова
(конечный список):



- значение было изменено.



- тэг и значение были изменены.

Тестирование лексического анализатора

Процессор и операционная система	Без оптимизации	С оптимизацией	Сокращение времени
Core i5 Windows 10	0.769 с	0.651 с	15.3 %
Core i7 Ubuntu 14.04	0.870 с	0.767 с	11.8 %

Заключение

Разработанный алгоритм успешно реализован на практике. Время выполнения оптимизированной программы сокращается на 8-14 процентов в зависимости от исходного кода программы.