

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Н. Э. БАУМАНА

ВЫЧИСЛИТЕЛЬ И ВЕРИФИКАТОР ТИПОВ ФУНКЦИЙ ДЛЯ ЯЗЫКА РЕФАЛ-5

Выполнил:
Иванов Георгий

Научный руководитель:
Коновалов Александр Владимирович

МОСКВА, 2019

Цель и задачи

- Целью данной работы является разработка верификатора типов в РЕФАЛ-5, который будет проверять корректность вызовов функций в программе
- Задачи – вывод и проверка типов для динамически типизированного языка РЕФАЛ-5

РЕФАЛ

РЕФАЛ (РЕкурсивных Функций АЛгоритмический язык) - это один из старейших функциональных языков программирования, который ориентирован на символьные преобразования. РЕФАЛ был впервые реализован в 1966 году в России Валентином Турчиным.

Основа РЕФАЛа состоит в сопоставлении с образцом (последовательность строк(-и) символов со сбалансированными скобками), поэтому легко реализовать символьную обработку.

Основные понятия РЕФАЛ

Данными РЕФАЛа являются объектные выражения
– строки, состоящие из символов и
сбалансированных круглых скобок

Например:

$$(A 'bc' (10\ 20))\ P\ R$$

Образцовые выражения – строки, состоящие из
символов, сбалансированных круглых скобок и
переменных.

$$(A\ s.\ 1\ s.\ A\ (e.\ X))\ t.\ Y\ R$$

Основные понятия РЕФАЛ

Жёсткое выражение – это образцовое выражение без открытых и повторных переменных, т. е. на одном уровне скобок может находиться не более чем одна е-переменная.

Подстановка – множество замен переменных в образцовом выражении некоторыми выражениями соответствующего типа:

$$\sigma = v_i \rightarrow P_i, i = \overline{1 \dots N}$$

Уточнение или сужение образца – операция замены переменных в образце.

Обобщение образца – операция, обратная к уточнению.

Тип функции

Мы будем искать аппроксимацию области определения и области значений функций в виде жёстких выражений.

Тип функции

Область определения функции определяется множеством значений, которые могут принимать образцы.

```
G {  
    B e.X = 2;  
    s.1 e.X = '*';  
}
```

Тип может быть описан как:

$$G s e = s$$

Тип функции

Область определения также ограничивается областями допустимых значений для переменных в образцах (определяются из областей определений вызываемых функций).

```
F {  
    A e.X = 1;  
    e.Y = <G e.Y>;  
}
```

$$F s e = s$$

```
G {  
    B e.X = 2;  
    s.1 e.X = '*';  
}
```

$$G s e = s$$

Тип функции

Вычисление множество допустимых значений для переменных требует решения уравнения методом унификации.

Пусть дана функция с форматом:

$$F\ s\ s = \dots$$

То тогда для вызова функции в виде:

$$\langle F\ e.X\ e.Y \rangle$$

Решения будут иметь вид:

$$\begin{cases} e.X \rightarrow \varepsilon, & e.Y \rightarrow s.1\ s.2 \\ e.X \rightarrow s.1, & e.Y \rightarrow s.2 \\ e.X \rightarrow s.1\ s.2, & e.Y \rightarrow \varepsilon \end{cases}$$

Расширение алгоритма на полный РЕФАЛ-5. Условия

Пусть дано предложение

$$Pat, R_1 : P_1 = Res;$$

Решаем уравнение из R_1 (уравнения для вызовов функций из R_1).

Получаем набор решений, подставляем в предложение.

$$Pat', R'_1 : P'_1 = Res';$$

Далее, решаем уравнение.

$$R'_1 : P'_1$$

Получаем набор сужений и присваиваний. Далее, подставляем в предложения.

$$Pat'', R''_1 : P''_1 = Res'';$$

После, используем алгоритм вывода для базисного РЕФАЛа.

Расширение алгоритма на полный РЕФАЛ-5. Блоки

Пусть дано предложение

$$Pat, R_1 : P_1, Res : \{ P_{11} = R_{11}; P_{21} = R_{21}; \}$$

Преобразуем его в предложения с условиями:

$$Pat, R_1 : P_1, Res : P_{11} = R_{11};$$

$$Pat, R_1 : P_1, Res : P_{21} = R_{21};$$

Далее, применяем алгоритм вывода для полного РЕФАЛа (условия).

Выводы

В рамках ВКР:

- Разработан и реализован алгоритм вывода типа функций, а также реализована проверка корректности вызовов функций в программе;
- Проведено тестирование на более, чем 50 тестах

В ходе написания работы был разработан верификатор, который был реализован на языке Python3.

При этом в дальнейшем планируется:

- модификация алгоритма обобщения
- оптимизация работы верификатора.

Как установить?

```
pip3 install refalchecker
```