

Использование отношения Хигмана-Крускала для прерывания рекурсивной специализации функций

А. А. Кошелев

МГТУ им. Н. Э. Баумана

Третье совместное рабочее совещание
ИПС имени А. К. Айламазяна РАН и МГТУ имени Н. Э. Баумана
по функциональному языку программирования Рефал

12 июня 2020 года

Постановка задачи

Устранить зацикливание компилятора языка Рефал-5λ, возникающее при специализации функций, с помощью отношения Хигмана-Крускала.

Для этого необходимо:

- изучить причины возникновения зацикливания и способы его прерывания;
- разработать и реализовать алгоритм применения отношения Хигмана-Крускала и алгоритм обобщения сигнатур;
- проверить корректность работы реализованных алгоритмов.

Специализация функций

$F(X, Y)$ – функция с параметрами X, Y .

A – одно из возможных значений параметра Y .

Тогда $F_A(X) = F(X, A)$ – специализация функции $F(X, Y)$ по параметру Y .

Параметры, по которым осуществляется специализация, называют *статическими*.

Предполагается, что построение специализации позволит:

- увеличить скорость работы программы;
- уменьшить объём требуемой памяти.



Специализация функций в языке Рефал-5λ

- В языке Рефал-5λ каждая функция принимает один аргумент.
- Поэтому был введён шаблон специализации, указывающий специализируемую функцию и её статические переменные.

Пример использования шаблона специализации

$\$SPEC\ F\ t.x\ t.Y;$

$\$SPEC$ – ключевое слово (specialization);

F – имя функции, которую необходимо специализировать;

$t.Y$ – статическая переменная;

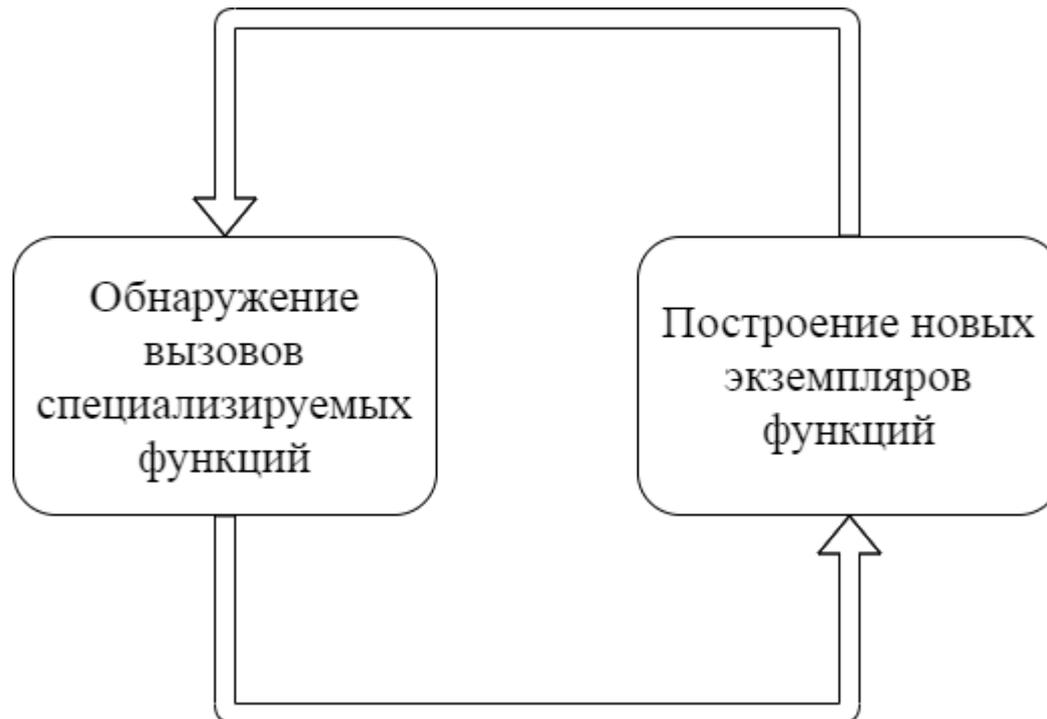
$t.x$ – динамическая переменная.

Алгоритм специализации функций

1. Происходит поиск вызова специализируемой функции.
2. Определяются значения статических переменных.
- 3.1. Если значения новые, строится новый экземпляр функции.
- 3.2. Иначе указывается вызов уже построенного экземпляра.
4. Запоминается сигнатура в виде
(имя экземпляра; значения статических переменных).

Зацикливание при специализации

В ряде случаев алгоритм специализации может зациклиться.

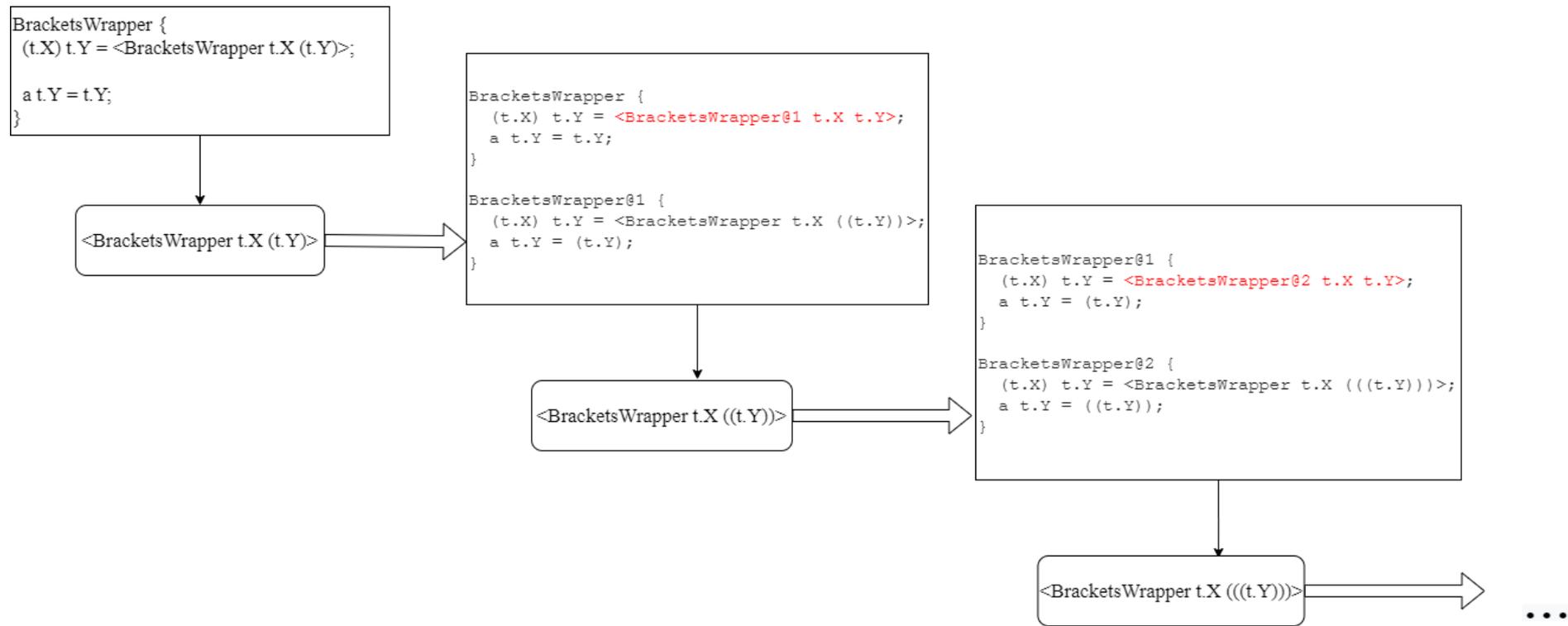


Пример зацикливания специализации

```
$SPEC BracketsWrapper t.x t.Y;
```

```
BracketsWrapper {  
  (t.X) t.Y = <BracketsWrapper t.X (t.Y)>;  
  
  a t.Y = t.Y;  
}
```

Процесс зацикливания специализации



Специализация данной функции с помощью приведённого алгоритма не может завершиться.

Обнаружение зацикливания

- Сравниваем сигнатуры с помощью отношения Хигмана-Крускала (обозначается знаком \leq).

- Интуитивное определение отношения Хигмана-Крускала:

для сигнатур S_1, S_2 выполняется $S_1 \leq S_2$, если путём удаления из S_2 некоторых элементов (переменных, скобочных символов, функций и т. д.) можно получить S_1 .

- Примеры:

$$x \leq (y); \quad ((y)) \leq (((y))); \quad x y \leq f(g(x), y)$$

Формальное определение

Отношение Хигмана-Крускала имеет индуктивное определение:

1. $x \leq y$ для любых переменных x и y .

2. $X \leq f(Y_1, Y_2, \dots, Y_n)$, если f — функция и выполняется условие

$\exists i: X \leq Y_i$.

3. $f(X_1, X_2, \dots, X_n) \leq f(Y_1, Y_2, \dots, Y_n)$, если f — функция и выполняется

$\forall i = 1, \dots, n \ X_i \leq Y_i$.

Теорема Хигмана-Крускала

Отношение Хигмана-Крускала является отношением хорошего предпорядка для выражений в конечном алфавите.

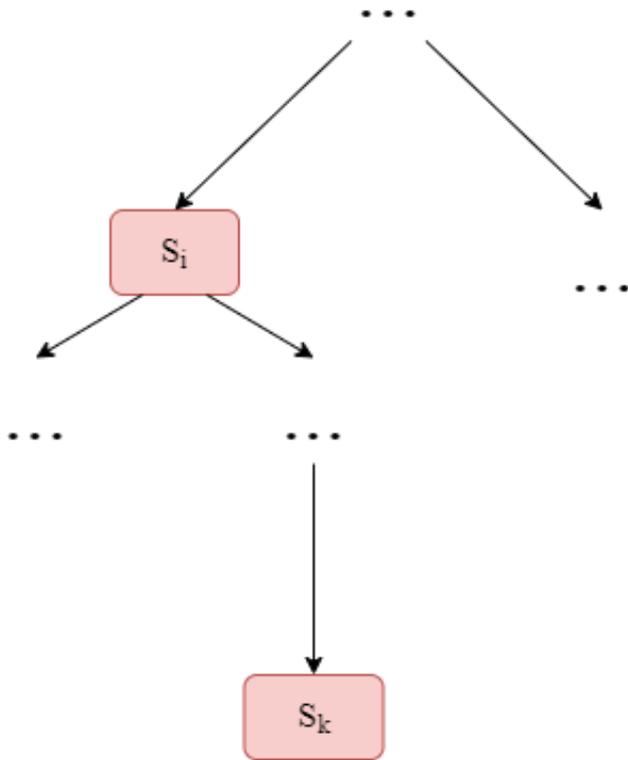
Для любой бесконечной последовательности сигнатур $\{S_i\}_{i \in \mathbb{N}}$, имена переменных в которых составлены из конечного алфавита:

$$\exists i, k: S_i \leq S_k$$

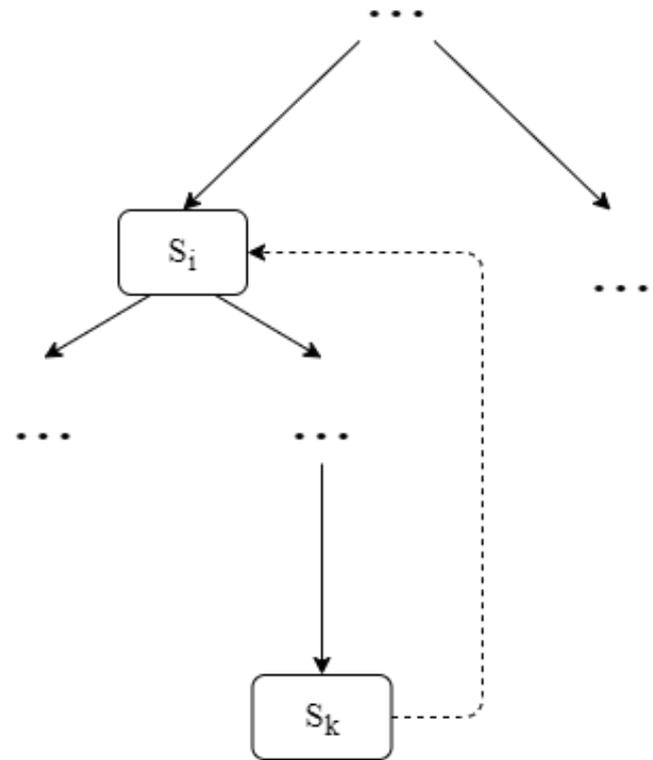
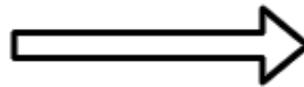
Обобщение сигнатур

- Пусть для сигнатур S_1, S_2 выполнилось $S_1 \leq S_2$.
- Обобщение S_1 и S_2 — сигнатура S_{gen} , из которой можно подстановками получить S_1, S_2 .
- Пример:
 $S_1 = () t. y, S_2 = ('a') t. y.$
 $S_{gen} = (e.x) t. y.$
При $e.x = \varepsilon$ получаем $S_{gen} \rightarrow S_1$.
При $e.x = 'a'$ получаем $S_{gen} \rightarrow S_2$.

Частный случай обобщения

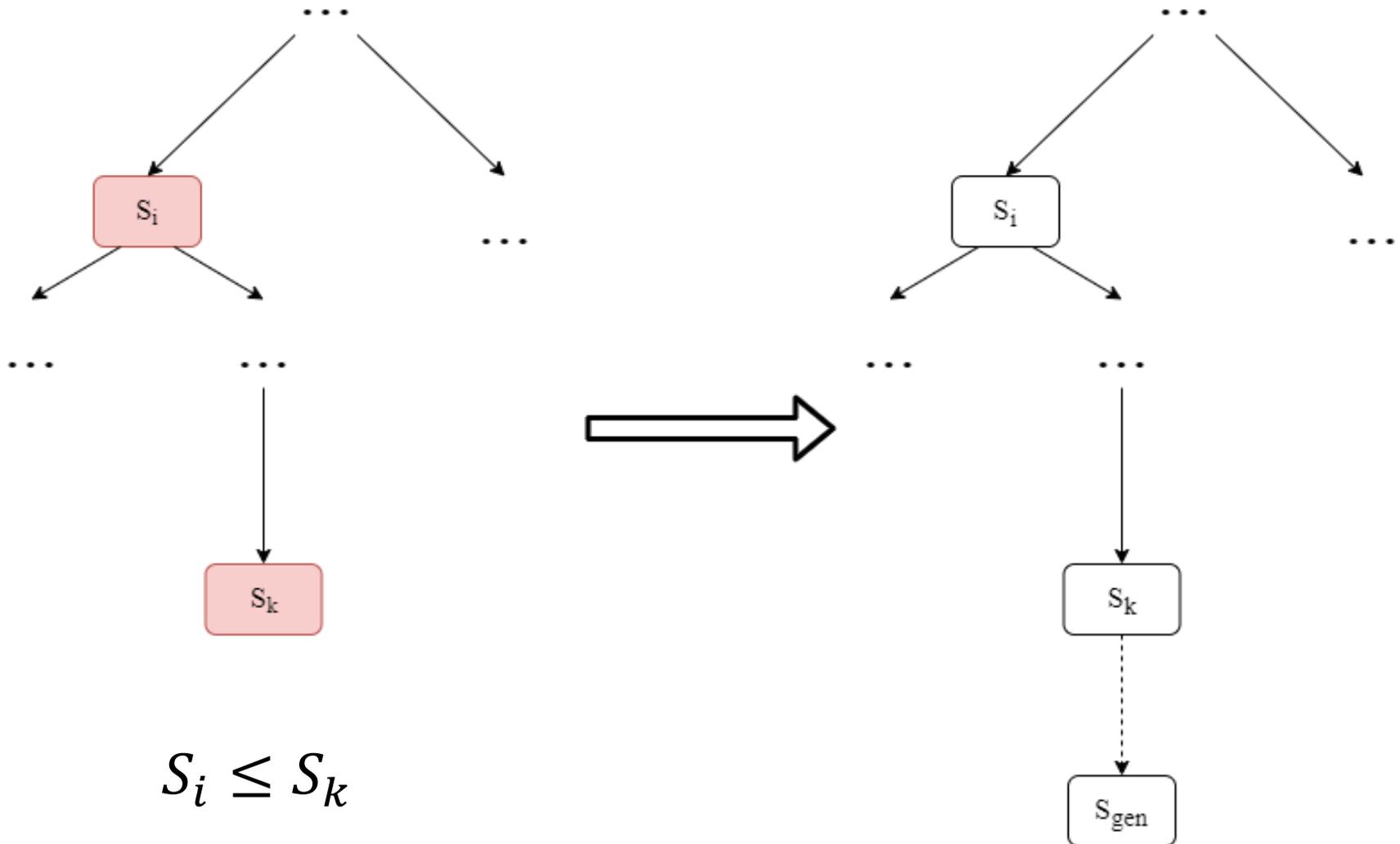


$$S_i \leq S_k$$



$$S_{gen} = S_i$$

Обобщение снизу



Реализация отношения Хигмана-Крускала

Пусть имеются выражения E_1 и E_2 , причём выражение E_2 можно представить в виде конкатенации $E_2'E_2''$. Тогда, если $E_1 \leq E_2$, то верно одно из трёх:

1. $E_1 \leq E_2'$.
2. $E_1 \leq E_2''$.
3. E_1 можно представить в виде конкатенации $E_1'E_1''$, где $|E_1'| > 0$ и $|E_1''| > 0$, для которой выполняются условия: $E_1' \leq E_2'$ и $E_1'' \leq E_2''$.

Реализация метода обобщения снизу

- Обобщение сигнатур в общем случае определяется неоднозначно.

- Пример:

$$S_1 = () t. y, S_2 = ('a') t. y$$

$$S_{gen_1} = (e. x) t. y \text{ или } S_{gen_2} = e. x$$

- Для сохранения максимального количества информации о сигнатуре использован алгоритм глобального сложнейшего обобщения.

Пример тестирования

До применения специализации

```
BracketsWrapper {  
  (t.X) t.Y = <BracketsWrapper t.X (t.Y)>;  
  
  a t.Y = t.Y;  
}
```

После применения специализации

```
BracketsWrapper {  
  (t.X) t.Y = <BracketsWrapper@1 t.X t.Y>;  
  
  a t.Y = t.Y;  
}  
  
BracketsWrapper@1 {  
  (t.X) t.Y = <BracketsWrapper@1 t.X (t.Y)>;  
  
  a t.Y = (t.Y);  
}
```

Заключение

- Было рассмотрено использование отношения Хигмана-Крускала и алгоритма обобщения снизу для решения проблемы зацикливания рекурсивной специализации.
- Методы были разработаны и реализованы в компиляторе языка Рефал-5λ.
- Тестирование показало корректность работы реализованных методов.