

# Использование отношения Хигмана-Крускала для прерывания рекурсивной специализации функций

Студент группы ИУ9-82: Кошелев А. А.  
Научный руководитель: Коновалов А. В.

# 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 и измерить изменение времени выполнения компиляции.

# Специализация функций

Пусть  $F$  – функция,  $F(args)$  – вызов функции  $F$ .  
*Специализация* – такое порождение новой функции  $F'$  и замена вызова  $F(args)$  на вызов  $F'(args')$ , что в  $F'$  учтена часть статически известной информации о её вызове (например, значения некоторых аргументов в  $args$ ).

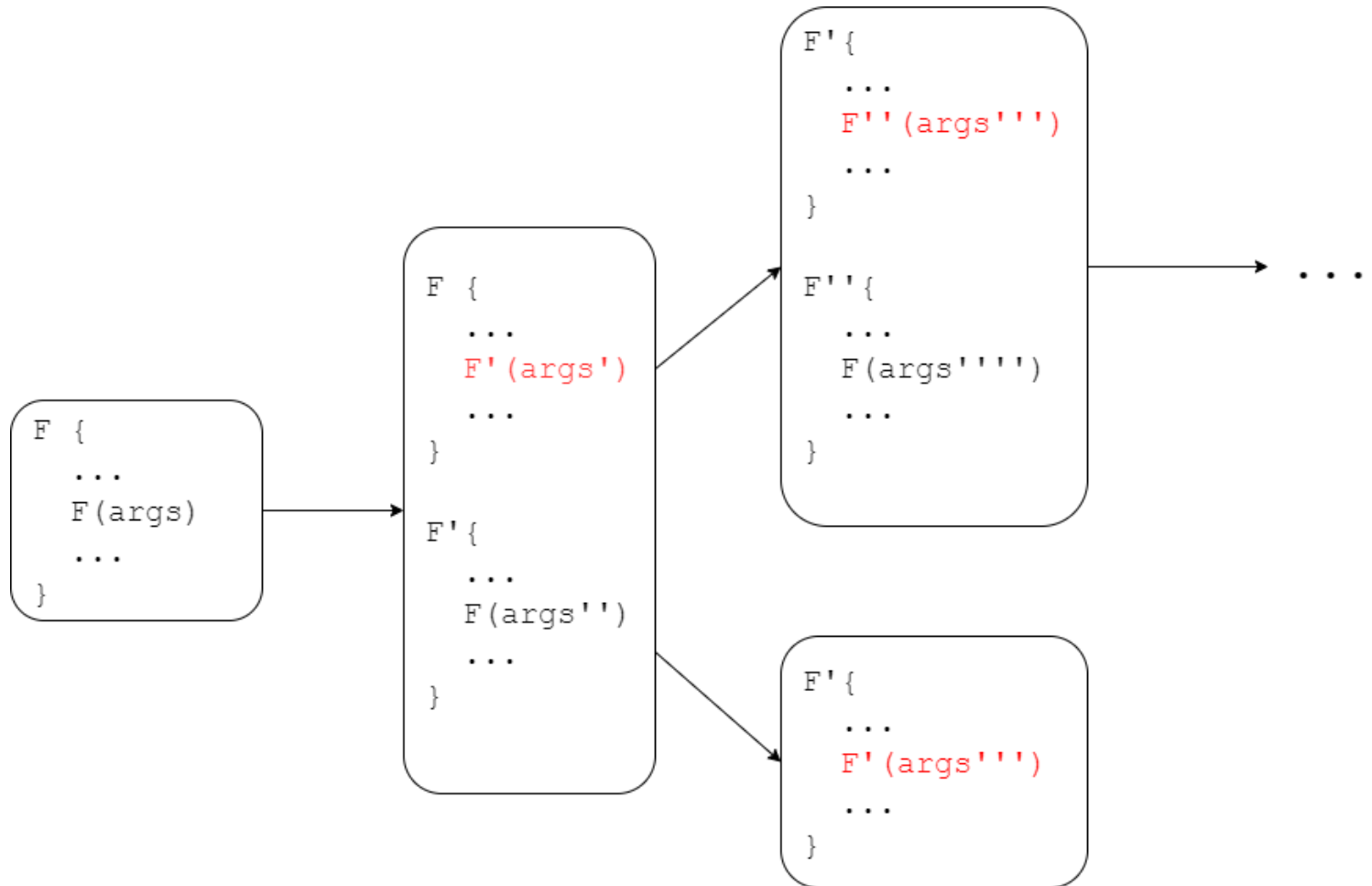
Тогда:

- $F'$  – экземпляр функции  $F$ ;
- информация, учтённая о вызове  $F(args)$  – *сигнатура* экземпляра  $F'$ .

# Алгоритм специализации функций

1. Происходит поиск вызова специализируемой функции.
2. Определяется сигнатура.
- 3.1. Если была получена уже встречавшаяся сигнатура, указывается вызов уже построенного экземпляра.
- 3.2. Если сигнатура новая, строится новый экземпляр функции, указывается её вызов, запоминается сигнатура.

# Специализация рекурсивных функций



# Зацикливание специализации



# Обнаружение зацикливания

- Сравниваем сигнатуры с помощью отношения Хигмана-Крускала (обозначается знаком  $\preceq$ ).
- Интуитивное определение отношения Хигмана-Крускала:  
для сигнатур  $S_1, S_2$  выполняется  $S_1 \preceq S_2$ , если путём удаления из  $S_2$  некоторых элементов (переменных, скобочных символов, функций и т. д.) можно получить  $S_1$  с точностью до переименования переменных.
- Примеры:

$$\begin{aligned}x &\preceq (x) \\ y\ z &\preceq f(g(x), y)\end{aligned}$$

# Формальное определение

Отношение Хигмана-Крускала имеет индуктивное определение:

1.  $x \sqsubseteq y$  для любых переменных  $x$  и  $y$ .
2.  $X \sqsubseteq f(Y_1, Y_2, \dots, Y_n)$ , если  $f$  — функция и  $\exists i: X \sqsubseteq Y_i$ .
3.  $f(X_1, X_2, \dots, X_n) \sqsubseteq f(Y_1, Y_2, \dots, Y_n)$ , если  $f$  — функция и  $\forall i = 1, \dots, n \ X_i \sqsubseteq Y_i$ .



# Теорема Хигмана-Крускала

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

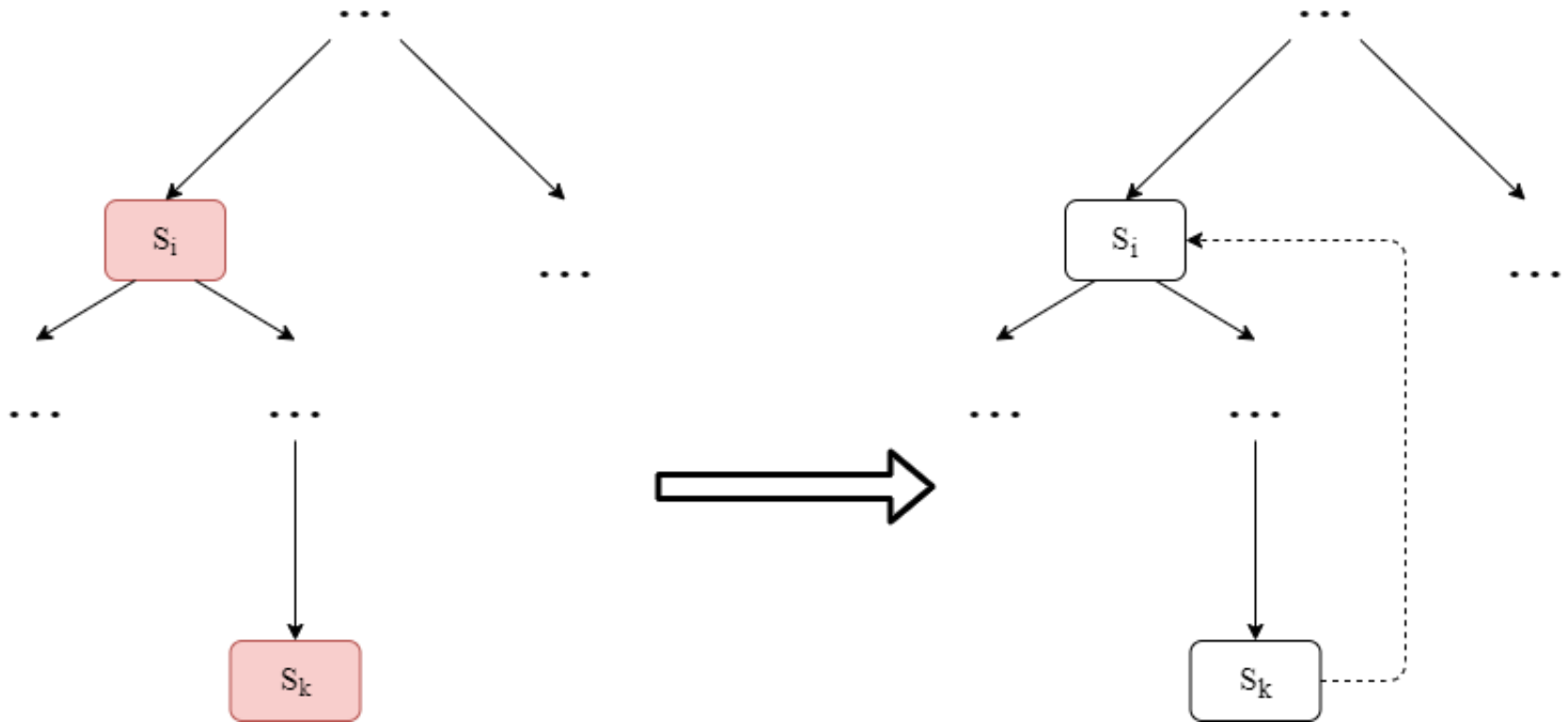
Для любой бесконечной последовательности сигнатур  $\{S_i\}_{i \in \mathbb{N}}$ , имена переменных в которых составлены из конечного алфавита:

$$\exists i, k \in \mathbb{N}, i < k: S_i \preceq S_k$$

# Обобщение сигнатур

- Пусть для сигнатур  $S_1, S_2$  выполнилось отношение Хигмана-Крускала:  $S_1 \preceq S_2$ .
- Обобщение  $S_1$  и  $S_2$  — сигнатура  $S_{gen}$ , из которой можно подстановками получить  $S_1, S_2$ .
- Пример:  
 $S_1 = ()y, S_2 = ('a')y.$   
 $S_{gen} = (x)y.$   
При  $x = \varepsilon$  получаем  $S_{gen} \rightarrow S_1$ .  
При  $x = 'a'$  получаем  $S_{gen} \rightarrow S_2$ .

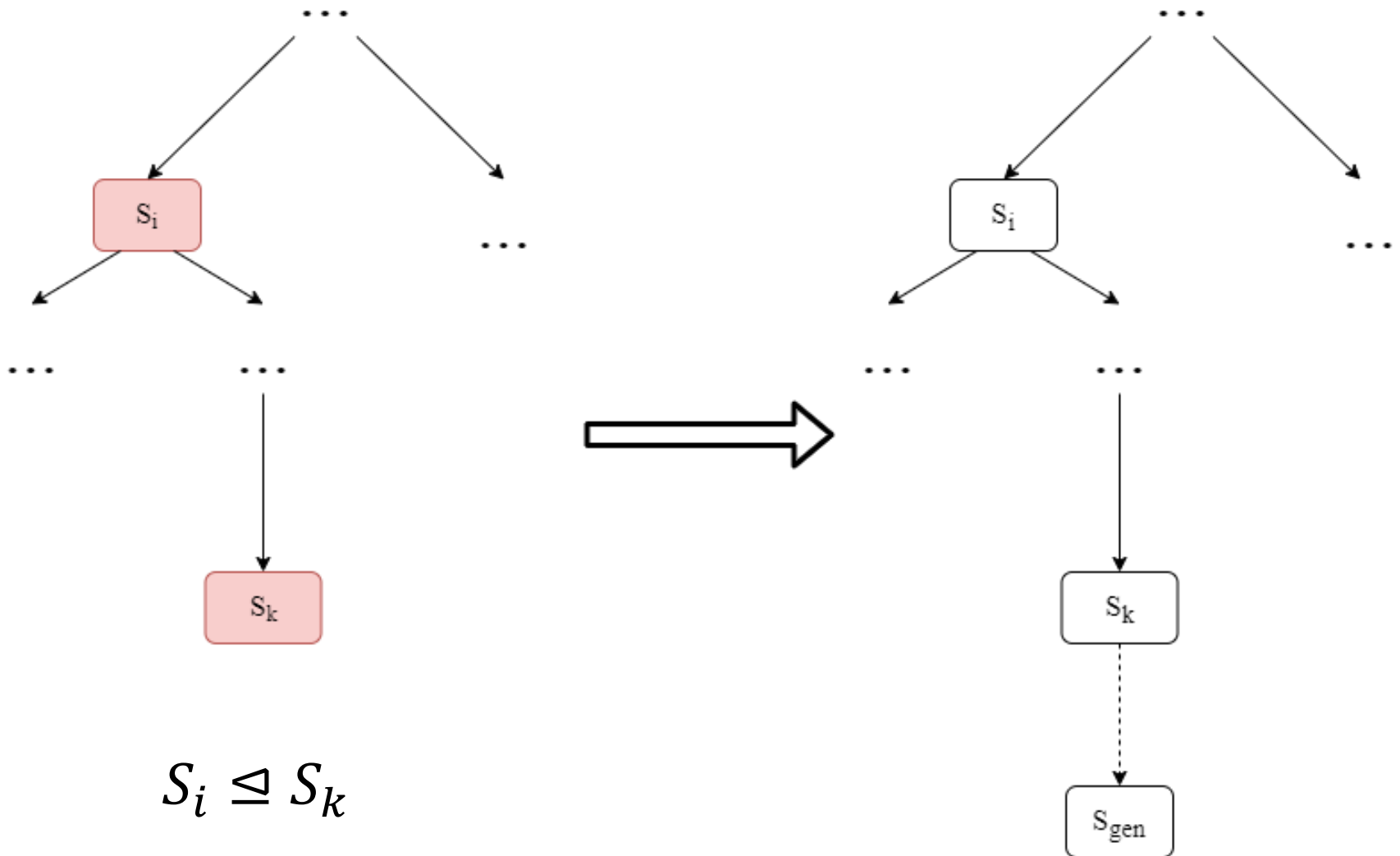
# Частный случай обобщения



$$S_i \trianglelefteq S_k$$

$$S_{gen} = S_i$$

# Обобщение снизу



# Специализация функций в языке Рефал-5λ

- Была реализована путём введением *статических* и *динамических* параметров функции.
- В теле экземпляра вместо всех вхождений статического параметра подставляется его фактическое значение (из аргумента).
- При этом значением параметра может быть выражение с переменными и вызовами функций.

# Разработка и реализация

- 1) Распознавание зацикливаний в цепочках экземпляров по отношению Хигмана-Крускала:
  - отслеживание истории сигнатур;
  - использование следствий из определения отношения для проверки его выполнения.
- 2) Обобщение похожих вызовов:
  - использование алгоритма глобального сложнейшего обобщения.

# Тестирование

- Проверена корректность работы алгоритмов.
- Проведены замеры времени компиляции исходного кода компилятора, в каждом случае 13 замеров:

Версия компилятора	Время компиляции, медиана, с	Время компиляции, I квартиль, с	Время компиляции, III квартиль, с
Без внесённых изменений	36.59	36.44	37.17
С внесёнными изменениями	36.97	36.83	37.09

- Медианное время увеличилось на 1%.
- Вывод: для программы, написанной в рекомендованном стиле программирования Рефала-5λ, логика проверки зацикливания не является узким местом.

# 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 и незначительное изменение времени выполнения компиляции.