

**IV-ое совместное рабочее совещание**  
**ИПС имени А. К. Айламазяна РАН**  
**и**  
**МГТУ имени Н. Э. Баумана**  
**по функциональному языку программирования Рефал**  
 (г. Москва, 8 июня, 2021 г.)

**Программа**

11:10	Открытие	
11:20	Первая сессия (председатель: Александр Коновалов)	
11:20 - 12:00	<b>О языке описания свойств параметризованных конфигураций (уравнение Хмелевского в словах) ( слайды )</b>	Андрей Немытых
12:05 - 12:30	<b>Построение решателя уравнений в словах методом развёртки/свёртки графа решений ( слайды )</b>	Александра Спиридонова
12:35 - 13:15	<b>Суперкомпиляция как основа для построения алгоритмов решения уравнений в словах ( слайды )</b>	Антонина Непейвода
13:15	Перерыв (обед)	
13:30	Вторая сессия (председатель: Антонина Непейвода)	
13:40 - 14:05	<b>Расширенный алгоритм обобщённого сопоставления в компиляторе Рефала-5λ ( слайды )</b>	Владислав Пичугин
14:10 - 14:35	<b>Алгоритм прогонки вызовов рекурсивных функций, не приводящий к заикливанию в компиляторе Рефала-5λ ( слайды )</b>	Михаил Апахов
14:50 - 15:30	<b>Декомпозиция вызовов функций во время суперкомпиляции путём построения выходных форматов ( слайды )</b>	Александр Коновалов
15:30	Перерыв (чай, кофе)	
16:05	Третья сессия (председатель: Андрей Немытых)	
16:00 - 16:40	<b>Проект №<sub>0</sub> – постановка задачи для создания платформонезависимой системы управления знаниями на базе языка Рефал</b>	Дмитрий Костин
16:40 - 17:05	<b>Типизация функций для Рефала-5 ( слайды )</b>	Дмитрий Сырбу
17:15 - 17:40	<b>О переборных алгоритмах проверки вложения конечнозначных структур ( слайды )</b>	Евгений Шевляков
17:40	Обсуждение	
17:55	Закрытие	

### Аннотации докладов

1. *Андрей Немытых, «О языке описания свойств параметризованных конфигураций (уравнение Хмелевского в словах)»*

В суперкомпиляторе MSCP-A реализован язык описания свойств параметризованных конфигураций  $\mathcal{L}$  специализируемой программы, основанный на языке уравнений в свободном моноиде. Необходимость в использовании языка  $\mathcal{L}$  следует из ассоциативности операции приписывания в языке Рефал. В докладе будут обсуждаться некоторые выразительные свойства языка  $\mathcal{L}$ .

2. *Александра Спиридонова, «Построение решателя уравнений в словах методом развёртки/свёртки графа решений»*

При суперкомпиляции программ возникает задача решения уравнений в словах.  $\mathcal{A}$  – алфавит констант;  $\mathcal{X}$  – алфавит строковых переменных;  $T_1 = T_2$  – уравнение в словах, где  $(T_1, T_2) \in (\mathcal{A} \cup \mathcal{X})^* \times (\mathcal{A} \cup \mathcal{X})^*$ .

Основная трудность, возникающая при их решении – отсутствие эффективного и полного алгоритма, который бы находил все решения уравнения. Реализованный алгоритм Матиясевича определяет, имеет ли уравнение в словах решение. Он основан на свёртке и развёртке дерева всех возможных решений, которое порождается некоторым «шагом прогонки», т. е. способом преобразования уравнения в множество уравнений – частных случаев исходного. Алгоритм использует для построения графа развёртки преобразования Нильсена.

Построенный решатель усиливает алгоритм Матиясевича при помощи анализа структуры уравнения в словах, определения принадлежности уравнения одному из специальных классов и последующего использования простых преобразований (конечные подстановки, противоречие по длинам, разбиение, упрощение).

3. *Антонина Непейвода, «Суперкомпиляция как основа для построения алгоритмов решения уравнений в словах»*

В докладе рассматривается способ использования суперкомпилятора в качестве каркаса для построения алгоритмов решения уравнений в словах. Для этого можно написать интерпретатор простого логического языка, проверяющего, приводит ли последовательность преобразований над уравнением к его решению, и затем проспециализировать такой интерпретатор конкретным уравнением, которое требуется решить. Если интерпретатор построен правильно, и специализация завершится, её результатом окажется описание всех решений уравнения. Ключевым свойством, обеспечивающим способность решать уравнения, является оптимальность пары

интерпретатор + суперкомпилятор

– соответствие операций свёртки и развёртки при специализации таковым в алгоритмах решения уравнений. Мы покажем, какие особенности рассмотренных в докладе интерпретаторов позволили добиться оптимальности специализации.

4. *Владислав Пичугин, «Расширенный алгоритм обобщённого сопоставления в компиляторе Рефала-5λ»*

Центральное место в языке программирования Рефал-5λ занимает механизм сопоставления с образцом, использующийся во время выполнения любой программы, а также в некоторых оптимизациях, происходящих на этапе компиляции (например, в таких оптимизациях как прогонка и специализация функций).

В общем случае алгоритм сопоставления имеет дело с уравнениями вида  $\mathbf{P} : \mathbf{L}$ , где  $\mathbf{P}$  и  $\mathbf{L}$  могут быть любыми образцовыми выражениями. В докладе рассказывается о том, как был реализован расширенный алгоритм обобщённого сопоставления, который позволяет единообразно получать решение таких уравнений в виде наборов сужений и присваиваний для всех граничных случаев, где это возможно, а также поддерживает механизм динамического обобщения, открывающий возможности для расширенной специализации функций.

5. *Михаил Апахов, «Алгоритм прогонки вызовов рекурсивных функций, не приводящий к заикливанию в компиляторе Рефала-5λ»*

Прогонка в существующей реализации выполняется по одному вызову функции за проход из-за того, что нет определения появления заикливания. Однако существует алгоритм, который позволяет определять заикливания и не прогонять повторно уже встречавшиеся выражения. Данный алгоритм основан на алгоритме ациклической суперкомпиляции – символического вычисления результата вызова функции без полного знания о значениях аргументов.

Будет рассказано о реализации данного алгоритма в Рефале-5λ. Такой подход позволит прогонять все функции, даже рекурсивные, при этом вызовы обычных функций должны прогоняться быстрее за счёт выполнения всех прогонок за один раз.

6. *Александр Коновалов, «Декомпозиция вызовов функций во время суперкомпиляции путём построения выходных форматов»*

Как известно, суперкомпиляция неплохо справляется с задачей вычисления композиции функций: вызов  $\langle \mathbf{F} \langle \mathbf{G} \mathbf{e.X} \rangle \rangle$  в остаточной программе преобразуется в вызов одной функции  $\langle \mathbf{Fg} \mathbf{e.X} \rangle$ , выражающей результат композиции исходных функций. Однако, простые методы суперкомпиляции не могут сделать обратное преобразование: преобразовать вызов функции исходной программы  $\langle \mathbf{F} \mathbf{e.X} \rangle$  в композицию двух вызовов остаточной программы  $\langle \mathbf{F}'' \langle \mathbf{F}' \mathbf{e.X} \rangle \rangle$ .

В докладе будет рассказано, как, строя во время суперкомпиляции выходные форматы функций, производить декомпозицию: разбивать вычисление одной функции на композицию двух.

7. *Дмитрий Костин, «Проект №0 – постановка задачи для создания платформонезависимой системы управления знаниями на базе языка Рефал»*

На социальном, некоммерческом подходе планируется создать прототип программного модуля, аналога даймона в Unix-подобных ОС со способом управления на основе передачи сигналов по гибридной схеме, технологии блокчейн и клиент-сервер. Вводится понятие пространства имен, в котором используются модели «владелец», «роль владельца», «время жизни программы».

В качестве базы предполагается использовать язык Рефал-05, суперкомпилятор SCP4. Операционные системы Plan9Front, OpenBSD, FreeBSD, Debian. Для Windows будет задействован эмулятор Qemu.

8. *Дмитрий Сырбу, «Типизация функций для Рефала-5»*

Введение в язык механизма типизации позволит проверять корректность, написанных программистом функций до этапа компиляции программы.

Верификация типов будет заключаться в сопоставлении фактических типов аргументов и возвращаемых значений с формальными, описанными в нотации типов.

Данный доклад посвящён разработке представления типов функций языка Рефал и процедуры автоматической верификации типов. В докладе будет представлено описание подмножества **КС**-грамматик, замкнутое относительно теоретико-множественных операций, операции конкатенации и сопоставления с рефал-образцами (значения переменных также описываются в этом подмножестве).

9. *Евгений Шевляков, «О переборных алгоритмах проверки вложения конечнозначных структур»*

В докладе будет рассказано, о реализации переборных алгоритмов, проверяющих функциональное вложение одной конечнозначной логики в другую.

Конечнозначные логики описываются наборами функций на простом рефале. С помощью некоторых оптимизаций удалось решить две задачи функционального вложения четырехзначных паралогик Левина-Микенберг.