

**V-ое совместное рабочее совещание
ИПС имени А.К. Айламазяна РАН
и
МГТУ имени Н.Э. Баумана
по функциональному языку программирования Рефал
(г. Москва, 16 июня, 2022 г.)**

Программа

12:00	Открытие	
12:05	Первая сессия (председатель: Антонина Непейвода)	
12:05–13:05	Об одном равномерном свойстве суперкомпилятора SCP4 (слайды)	Андрей Немытых
13:10–14:30	Чистый рекурсивный спуск на Рефале-5 (слайды)	Александр Коновалов
14:30–15:00	Обед	
15:00	Вторая сессия (председатель: Александр Коновалов)	
15:00–15:40	Обобщение при уточнении понятия закрытой переменной (слайды)	Антонина Непейвода
15:45–16:25	Типизация функций подмножества языка РЕФАЛ (слайды)	Дмитрий Сырбу
16:30	Заккрытие	

Аннотации докладов

1. Андрей Немытых, «Об одном равномерном свойстве суперкомпилятора SCP4» (построение специализированных версий алгоритма Матиясевича посредством специализации наивного поиска образца в строке)

В 1988 году Futamura и Nogi опубликовали результаты нескольких экспериментов по специализации программной модели наивного поиска образца в строке по конкретным образцам. Структура остаточных программ, автоматически построенных методом обобщённых частичных вычислений (*generalized partial computation*), совпала со структурой специализированных версий алгоритма Матиясевича, по недоразумению часто называемого алгоритмом Кнута-Морриса-Пратта. Таблица переходов по строке, которую строит первая стадия алгоритма Матиясевича, была автоматически вставлена в остаточные программы.

Результаты Futamura-ы и Nogi существенно опираются на использовании методом обобщённых частичных вычислений «отрицательной информации» о параметризованных конфигурациях специализируемой программной модели, которая была определена в терминах хвостовой рекурсии.

Позже подобные эксперименты повторялись многими авторами в разных контекстах для демонстрации возможностей разных модельных специализаторов.

В 1989 году R.S. Bird, J. Gibbons и G. Jones представили в терминах функционального языка высшего порядка ручной просчёт преобразования модели наивного в общем положении поиска образца в строке, где образец неизвестный — произвольный — параметризованный. Результатом их вычислений является алгоритм Матиясевича в общем положении. Метод преобразования основан на алгебраической технике и использует не автоматизированные трюки, включая отношения высшего порядка понижающие сложность вычисления в худшем случае преобразуемой программной модели.

В докладе будет рассказано о ключевых свойствах суперкомпилятора SCP4, которые позволяют ему преобразовать для любого фиксированного образца программную модель наивного поиска, определённую в терминах хвостовой рекурсии, к соответствующей, специализированной по этому образцу, версии алгоритма Матиясевича. Описанное здесь свойство суперкомпилятора SCP4 невозможно проверить посредством экспериментов, поскольку существует бесконечно много образцов. В докладе будут также сформулированы основные утверждения, из которых следует это равномерное по множеству образцов свойство суперкомпилятора SCP4.

2. *Александр Коновалов, «Чистый рекурсивный спуск на Рефале-5»*

Рекурсивный спуск — хорошо известная методика написания парсеров вручную. Методика позволяет, имея LL(1)-грамматику в виде БНФ или РБНФ, написать регулярным образом заготовку кода синтаксического анализатора, которую затем можно наполнять семантическими действиями.

Классический рекурсивный спуск императивен. Он подразумевает наличие глобальной переменной `CurrentToken`, в которой хранится текущий токен, и функции с побочным эффектом `NextToken`, которая считывает новый токен и обновляет содержимое этой переменной.

В докладе будет рассказано, как перенести методику рекурсивного спуска на Рефал-5 без побочных эффектов (т.е. без изменяемых ящиков и без копилки). Т.е. как можно, имея грамматику в БНФ, регулярным образом написать заготовку парсера, которую затем можно будет наполнять семантическими действиями.

Прямое следование этой методике приводит к написанию неэффективного кода, который просится быть оптимизированным во время компиляции. Будет показано, что этот код можно будет оптимизировать при помощи суперкомпиляции.

3. *Антонина Непейвода, «Обобщение при уточнении понятия закрытой переменной»*

Не все синтаксически открытые переменные в образцах действительно требуют неоднозначных подстановок. Наличие структурных скобок или отрицательных рестрикций в некоторых случаях позволяет рассматривать кортеж образцов вместо единственного образца, в связи с чем иногда значения всех переменных, входящих в такие кортежи, определяется однозначно.

Мы рассмотрим простое расширение понятия закрытой переменной, которое позволяет строить более точные обобщения конфигураций, содержащие несколько подряд идущих -параметров, и приведём набросок доказательства нётеровости обобщаемых образцов относительно (нестроого) включения их языков при условии, что переменные в данных образцах являются обобщённо закрытыми.

4. *Дмитрий Сырбу, «Типизация функций подмножества языка РЕФАЛ»*

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. Нотация типов представляет регулярную грамматику, что позволяет выражать типы как регулярные выражения, а значит использовать для сопоставления эквивалентные конечные автоматы.

Доклад посвящён разработке представления типов функций языка РЕФАЛ и процедуры автоматической верификации типов.