

ного построения множественных выравниваний структур белков.
Сборник трудов конференции ПАВТ'2016, С. 81–92.

О РАСШИРЕНИИ ВОЗМОЖНОСТЕЙ ОДНОЙ АБСТРАКТНОЙ ЛИСП-МАШИНЫ

Работа удостоена диплома III степени

Галкина Екатерина Владимировна

Кафедра алгоритмических языков

e-mail: catherine.galkina@gmail.com

***Научный руководитель: к.ф.-м.н., доц. Столяров Андрей
Викторович***

Работа посвящена расширению функциональности вычислительной модели диалектов языка Лисп (IntelLib Lisp и IntelLib Scheme), представленной в библиотеке IntelLib, реализующей метод непосредственной интеграции [1]. Указанный метод является одним из способов многостилевого программирования. В его основе лежит моделирование средствами заранее выбранного базового языка (в данном случае используется язык C++) синтаксиса и семантики альтернативных языков.

Вычисление S-выражений основывается на работе с континуацией [2]. Континуация представляет собой последовательность действий, которые необходимо выполнить до завершения вычислений. Рассматриваемая вычислительная модель представляет собой трёхстековую виртуальную машину, состоящую из стека действий, стека результатов и стека фреймов, используемого для оптимизации передачи параметров в функции.

В предыдущих версиях библиотеки IntelLib лексическое связывание реализовывалось с помощью однонаправленных списков соответствий между символом и его текущим значением. Для вычисления значения символа производился линейный поиск по этому списку. В работе предложен новый механизм лексического связывания, основанный на использовании позиционных параметров — специальных символов, предоставляющих прямой доступ к нужному значению параметра в стеке. Замена символов на позиционные параметры происходит при создании лексического замыкания, а именно при использовании λ -выражений и форм с предварительной инициализацией. В теле форм заменяются те символы, которые присутствуют в списке формальных параметров. Для реализации указанной замены был разработан специальный класс, инкапсулирующий контекст замены и предоставляющий методы для обработки атомов и различных списочных структур. Если выражение представляет собой вызов некоторой спецформы, то для дальнейшей обработки вызывается метод, отдельно определённый для каждой из них и зависящий от семантики формы.

В диалекте IntelLib Lisp есть возможность объявлять динамически связываемые переменные [3]. Их значение определяется во время вычислений,

в то время как значение статической переменной зависит от контекста места её определения. В предложенной реализации динамические символы не заменяются на позиционные параметры, связь со значением устанавливается напрямую. Кроме того, перед вычислением формы, в которой устанавливается локальная связь, планируется восстановление предыдущего значения, используя команды вычислителя.

В оба диалекта были добавлены функции и формы для совершения нелокальных динамических переходов. Их реализация основывается на использовании стека фреймов, входящего в состав вычислителя. Структуры, из которых он состоит, хранят позиции стеков действий и результатов, которые нужно восстановить при совершении перехода. Кроме того, в них сохраняется дополнительная информация (формы очистки и метка формы SATCN). Поиск в стеке осуществляется при генерации исключения в диалекте Intelib Lisp или при замене континуации в диалекте Intelib Scheme.

Кроме того, указанные диалекты были снабжены функциями и формами для возврата нескольких значений. Общая схема их работы состоит в следующем: значения генерируются в одной или нескольких формах и передаются в качестве аргументов принимающей функции. Поскольку в ходе обычных вычислений континуация принимает только одно значение [4], была добавлена команда для указания возможности возврата нескольких значений. Её аргумент — количество уже сгенерированных на данный момент значений. При их генерации в специальных формах по возможности в стек результатов помещаются все полученные значения, и их общее количество обновляется. Если континуация не принимает несколько значений, то возвращается только первое из них. Подсчёт общего числа сгенерированных значений необходим для того, чтобы корректно вызвать принимающую функцию с нужным числом аргументов.

В ходе работы были получены следующие результаты: в рамках библиотеки Intelib была предложена новая реализация лексических контекстов и динамического связывания переменных, исключая линейный поиск при вычислении их значений, благодаря чему эффективность вычислений увеличивается на 30–50%; диалекты Intelib Lisp и Intelib Scheme были снабжены функциями и спецформами для осуществления динамических нелокальных переходов, возврата нескольких значений и использования многозначных функций.

Литература

1. И. Г. Головин, А. В. Столяров. *Объектно-ориентированный подход к мультипарадигмальному программированию*. Вестник МГУ, сер.15 (ВМиК). 2002. № 1. С. 46–50.
2. А. В. Столяров. *Импорт вычислительной модели языка Scheme в объектно-ориентированное окружение*. Сборник статей молодых

учёных факультета ВМК МГУ. 2008. № 5. С. 119–130.

3. Э. Хювёнен, И. Сеппянен. *Мир Лиспа. В 2-х т.* М.: Мир, 1990.
4. R. Kelsey, W. Clinger, J. Rees. (eds.). *Revised⁵ Report on the Algorithmic Language Scheme*. ACM SIGPLAN Notices. 1998. 33. № 9. P. 26–76.

МЕТОД НИСХОДЯЩЕГО СИНТАКСИЧЕСКОГО АНАЛИЗА НА ОСНОВЕ L-ГРАФОВ

Кондратьев Григорий Дмитриевич

Кафедра алгоритмических языков

e-mail: grigoree@list.ru

*Научный руководитель: к.ф.-м.н., старший преподаватель
Алексей Александрович Вылиток*

Бесконтекстные языки являются важным классом формальных языков. Они широко используются, например, при описании синтаксиса языков программирования и структурированных данных [1].

Хорошо известна характеристика бесконтекстных языков бесконтекстными грамматиками. Существует другой способ описать языки этого класса – с помощью L-графов [2, 3]. Этот способ является развитием подхода, описанного Л. И. Станевичене в [4], где предложены новые, более прозрачные решения классических задач теории формальных языков.

В данной работе показано, что графовые представления также позволяют более просто объяснить некоторые классические методы синтаксического анализа. Предложен оригинальный алгоритм построения L-графа по КС-грамматике, корректность алгоритма доказана. На основе этого алгоритма описан метод нисходящего синтаксического анализа по L-графу, а также показана возможность определения применимости метода рекурсивного спуска с помощью L-графа.

Предложенные алгоритмы реализованы в виде программной системы на языке Python.

Литература

1. Ахо А., Ульман Дж. *Теория синтаксического анализа, перевода и компиляции, том 1* – М.: Мир, 1978. – 613 с.
2. Вылиток А. А., Ростовский А. В. *О подклассах графовых представлений формальных языков // Сборник тезисов посвященных 300-летию М.В. Ломоносова: "Ломоносовские чтения 2011 года.* – 2011. – с. 98-99.
3. Вылиток А. А., Сутырин П. Г. *Характеризация формальных языков графами // Сборник тезисов научной конференции "Тихоновские чтения Москва, МГУ имени М.В.Ломоносова, факультет ВМК, 25-29 октября 2010 г.* – 2010.