Ostap: синтаксическое расширение OCaml для создания парсер-комбинаторов с поддержкой левой рекурсии

Вербицкая Е. А., kajigor@gmail.com
Санкт-Петербургский государственный университет,
Россия, 199034, Санкт-Петербург, Университетская наб. 7/9

7 февраля 2017 г.

Аннотация

Парсер-комбинаторы — привлекательная техника реализации синтаксических анализаторов. Одним из недостатков подхода является невозможность использования левой рекурсии при объявлении парсер-комбинаторов. В докладе будет представлено синтаксическое расширение языка OCaml для создания парсер-комбинаторов, позволяющее реализовывать в том числе леворекурсивные парсер-комбинаторы.

Ключевые слова: парсер-комбинаторы, OCaml, левая рекурсия

Одним из способов реализации синтаксического анализа (проверки принадлежности предложения заданному языку и построения для него некоторого структурного представления) является техника парсер-комбинаторов [2]. Основная идея подхода заключается в моделировании синтаксических анализаторов фукнциями высшего порядка и комбинирования их при помощи небольшого множества комбинаторов. Существует множество разничных стилей реализации подхода, среди которых особо выделяются монадические парсер-комбинаторы [3], позволяющие осуществлять синтаксический анализ, зависящий от контекста, что может быть полезно при разборе, например, двумерного синтаксиса. Помимо способности разбирать более широкий класс языков, чем класс контекстно-свободных, парсер-комбинаторы также предоставляют возможность вычислять семантические значения, не строя абстрактное синтаксическое дерево. Другим существенным преимуществом является возможность создавать парсеры высшего порядка, то есть синтаксические анализаторы, которые принимают аргументом другие парсеры. Данная особенность повышает композиционность и сокращает размер реализации синтаксического анализатора.

Библиотека Ostap предоставляет набор парсер-комбинаторов и синтаксическое расширение языка OCaml для упрощения реализации синтаксических анализаторов. Библиотека реализует монадические парсер-комбинаторы с неограниченным предпросмотром, при этом коминатор выбора игнорирует вторую альтернативу, если анализ с помощью первой завершился успешно. Таким образом, результатом анализа всегда является единственный возможный вывод данного предложения (если он существует).

Парсер-комбинаторы Ostap полиморфны относительно типа входного потока. Данная особенность позволяет, с одной стороны, унифицированно анализировать потоки символов из разных источников: будь то поток стандартного ввода, файл или данные, пришедшие по сети. С другой стороны, становится возможным реализовывать дополнительную функциональность в объекте, моделирующем вход: например, производить лексический анализ (выделние лексем). В библиотеке есть стандартная реализация потока Matcher, инициализируемая строкой, производящая базовый лексический анализ и содержащая информацию о координатах анализируемой лексемы.

Одним из недостатков нисходящего синтаксического анализа, реализацией идей которого являются парсер-комбинаторы, является невозможность обработки леворекурсивных определений парсеров. Леворекурсивные правила являются наиболее естественным способом описать левоассоциативные операции, поэтому их поддержка может значительно упростить создание синтаксических анализаторов. Существует несколько решений описанной проблемы [1; 6], ни одно из которых не в состоянии обрабатывать леворекурсивные парсеры высшего порядка. Библиотека парсер-комбинаторов Meerkat позволяет использовать левую рекурсию при создании анализаторов, однако достигается это путем явного построения леса разбора для данной строки [4].

Для поддержки левой рекурсии в библиотеке Ostap мы использовали идеи, применённые для её добавления в формализм Parser Expression Grammar [5]. Данный подход оперирует понятием ограниченной рекурсии: такое использование нетерминала в выводе, что количество леворекурсивных вызовов в нем ограничено некоторой константой. Если строка имеет вывод в данной грамматике, то количество вызовов каждого леворекурсивного нетерминала при разборе конечно, поэтому для обработки леворекурсивного нетерминала достаточно найти оптимальное ограничение левой рекурсии, что и осуществляется динамически во время процесса вывода. Во время поиска промежуточные данные сохраняются в таблицу мемоизации, которая должна быть глобальна для всех используемых парсер-комбинаторов, поэтому эта функциональность реализована в абстракции входного потока, являющейся наследником класса Matcher. Для использования левой рекурсии в случае парсеров высшего порядка в библиотеке предусмотрен парсер-комбинатор fix. К сожалению, производительность решения на данный момент в случае использования взаимной рекурсии не является удовлетворительной и ее улучшение является задачей на будущее.

Список литературы

1. *Frost R. A.*, *Hafiz R.*, *Callaghan P.* Parser combinators for ambiguous left-recursive grammars // International Symposium on Practical Aspects of Declarative Languages. — Springer. 2008. — С. 167—181.
2. *Hutton G.* Higher-order functions for parsing // Journal of functional programming. — 1992. — Т. 2, № 03. — С. 323—343.
3. *Hutton G.*, *Meijer E.* Monadic parser combinators: Technical Report / University of Nottingham. — University of Nottingham, 1996. —.

4. *Izmaylova A.*, *Afroozeh A.*, *Storm T. v. d.* Practical, General Parser Combinators // Proceedings of the 2016 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation. — St. Petersburg, FL, USA : ACM, 2016. — С. 1—12. — (PEPM '16). — ISBN 978-1-4503-4097-7. — DOI: 10.1145/2847538.2847539.
5. *Medeiros S.*, *Mascarenhas F.*, *Ierusalimschy R.* Left Recursion in Parsing Expression Grammars // Programming Languages: 16th Brazilian Symposium, SBLP 2012, Natal, Brazil, September 23-28, 2012. Proceedings / под ред. F. H. de Carvalho Junior, L. S. Barbosa. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. — С. 27—41. — ISBN 978-3-642-33182-4. — DOI: 10.1007/978-3-642-33182-4_4.
6. *Warth A.*, *Douglass J. R.*, *Millstein T. D.* Packrat parsers can support left recursion. // PEPM. — 2008. — Т. 8. — С. 103—110.