Применение парсер-комбинаторов для разбора булевых грамматик

Курбатова З.И., студент кафедры системного программирования СПбГУ,
zarina.kurbatova@gmail.com
Подкопаев А.В., асп. кафедры системного программирования СПбГУ,
anton@podkopaev.net

Аннотация

Булевы грамматики - расширение контекстно-свободных грамматик, позволяющее использовать конъюнкцию и отрицание в правых частях правил вывода. С их помощью можно описать более широкий класс языков. Популярным подходом к описанию синтаксических анализаторов является парсер-комбинаторная техника. Основным недостатком этой техники является невозможность использования левой рекурсии. В данной работе представлен подход к реализации парсер-комбинаторной библиотеки с поддержкой булевых грамматик, в том числе содержащих леворекурсивные правила.

Введение

Многие языки имеют строгую синтаксическую структуру, которую можно успешно описать с использованием формальных грамматик. Согласно иерархии Н.Хомского [1], формальные грамматики делятся на 4 вида: неограниченные, контекстно-свободные, контекстно-зависимые и регулярные. Как известно, с помощью контекстно-свободных грамматик может быть описан весьма узкий класс языков. Существует расширение контекстно-свободных грамматик – булевы грамматики, предложенные А.Охотиным [2]. Булевы грамматики более выразительны, поскольку в правых частях правил вывода появляется возможность использовать три основные логические связки: дизъюнкцию, конъюнкцию и отрицание.

Задачу синтаксического анализа можно сформулировать как задачу определения принадлежности слова некоторому языку. Иными словами, синтаксический анализатор принимает на вход строку и определяет, может ли строка порождаться грамматикой задаваемого языка. Одним из популярных подходов к реализации синтаксических анализаторов является парсер-комбинаторная техника: анализаторы представляются функциями высших порядков [3]. Парсер-комбинаторы – это функции высшего порядка, принимающие в качестве параметров анализаторы и возвращающие в качестве результата анализатор. Анализатор, в свою очередь, принимает в качестве входа строку и возвращает

некоторый результат: например, список пар, состоящих из обработанного элемента и оставшейся части строки. При данном подходе определяются базовые комбинаторы, а затем с их помощью определяются более сложные. Например, в качестве базовых комбинаторов можно использовать комбинатор последовательного применения и комбинатор альтернативного применения. Основной проблемой подхода является невозможность использования левой рекурсии. Существуют различные реализации данной техники, поддерживающие левую рекурсию. Например, в работе [4] решение основано на алгоритме обобщенного синтаксического анализа GLL.

Постановка задачи

Целью данной работы является изучение применимости парсер-комбинаторной техники для разбора булевых грамматик. Для ее осуществления были поставлены следующие задачи:

- реализация эффективной парсер-комбинаторной библиотеки с поддержкой левой рекурсии на языке Kotlin;
- апробация библиотеки;

Булевы грамматики

На практике часто встречаются языки, которые нельзя описать с помощью контекстно-свободных грамматик. Например, рассмотрим язык $\{\omega\omega|\omega \in a, b^*\}$, которой не является контекстно-свободным. Его можно задать с помощью булевой грамматики:

$$S \to \neg AB\&\neg BA\&C$$
$$A \to XAX|a$$
$$B \to XBX|b$$
$$C \to XXC|\varepsilon$$
$$X \to a|b$$

Рис. 1: Грамматика языка $\{\omega\omega|\omega \in a, b^*\}$

Булева грамматика - это четверка $G = (\Sigma, N, P, S)$, где:

- Σ - конечное непустое множество терминалов;
- N - конечное непустое множество нетерминалов;
- P - конечное множество правил;

Каждое правило имеет следующий вид:

$$A \to \alpha_1 \& ... \& \alpha_m \& \neg\beta_1 \& ... \& \neg\beta_n$$

$$(m + n \geq 1, \alpha_i, \beta_i \in (\Sigma \cup N)^*)$$

Правило A можно интерпретировать следующим образом: если строка представима в форме $\alpha_1, ..., \alpha_m$, но не представима в форме $\beta_1., ..., \beta_n$, то она выводится из нетерминала A. Булевы грамматики расширяют контекстно-свободные грамматики, позволяя использовать в правых частях правил вывода конъюнкцию и отрицание.

Описание подхода

Для поиска всех возможных выводов строки используется техника программирования в стиле передачи продолжений (Continuation Passing Style). Идея техники заключается в передаче управления через механизм продолжений. Продолжение представляет собой состояние программы, в которое может быть осуществлен переход из любой точки программы. В нашем случае анализатор передает результат работы продолжению, которое представляет собой следующий этап синтаксического анализа. Анализ производится вне зависимости от порядка альтернатив, таким образом обеспечивается исчерпывающий поиск.

В библиотеке реализованы четыре базовых анализатора: eps для обработки пустого символа, term для обработки терминального символа, seq для последовательного применения и rule для альтернативного применения. На их основе можно конструировать более сложные анализаторы.

Для поддержки леворекурсивных правил используется техника мемоизации. Идея заключается в сохранении результатов анализа: если анализатор уже запускался на i-ой позиции, то результат берется из таблицы, лишь в противном случае происходит вычисление. Таким образом исключаются повторные вычисления. Более того, для избежания экспоненциальной сложности созданные во время анализа продолжения также мемоизируются.

Для поддержки булевых грамматик необходимо добавить два новых анализатора: andp для проверки принадлежности строки двум языкам и notp для проверки непринадлежности строки языку. Во-первых, нужно уметь определять, принадлежит ли строка пересечению двух языков, а во-вторых, уметь определять, принадлежит ли строка одному языку, а другому - нет. Например, язык $\{a^n b^b c^n\}$ можно задать с помощью конъюнктивной грамматики как

пересечение языков $\{a^i b^j c^k | i = j\}$ и $\{a^i b^j c^k | j = k\}$: строка принадлежит исходному языку, если принадлежит обоим языкам. На данный момент в библиотеке реализована поддержка конъюнктивных грамматик.

Заключение

В данной работе представлен подход к реализации парсер-комбинаторной библиотеки с поддержкой конъюнктивных грамматик, содержащих леворекурсивные правила. Подход основан на комбинировании двух техник: программирование в стиле передачи продолжений и мемоизация. В дальнейшем планируется добавление поддержки булевых грамматик и апробация.

Литература

[1] Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. "Automata theory, languages, and computation." International Edition 24 (2006).

[2] Okhotin, Alexander. "Boolean grammars." Information and Computation 194.1 (2004): 19-48.

[3] Hutton, Graham, and Erik Meijer. "Monadic parser combinators." (1996).

[4] Spiewak, Daniel. "Generalized Parser Combinators." (2010).