Санкт-Петербургский государственный университет

Кафедра Системного программирования

Ершов Кирилл Максимович

Синтаксический анализ графов с помеченными вершинами и ребрами

Курсовая работа

Научный руководитель:
ст. преп., к. ф.-м. н. Григорьев С. В.

Санкт-Петербург
2017

Оглавление

Введение

Помеченные графы являются удобным способом представления различных структурированных данных. Такие графы используются, например, в биоинформатике, логистике, графовых базах данных.

Иногда для представления данных с использованием графов обходятся только метками на рёбрах. Но в некоторых случаях метки на вершинах позволяют более наглядно отображать зависимости между сущностями. К примеру, в биоинформатике существует большое количество данных, содержащих взаимосвязь между генами и белками. Такие данные удобно представлять в виде графа, вершины которого помечены определенными генами и белками, а ребра показывают их отношение (например, ген кодирует белок).

Для эффективной работы с помеченными графами необходимо иметь возможность делать запросы, возвращающие нужную информацию из графа. Запросы можно представлять в виде грамматики. Тогда язык грамматики задает класс путей, удовлетворяющих запросу. Пути рассматриваются как строки, состоящие из меток на рёбрах и вершинах. Путь удовлетворяет запросу, если строка принадлежит соответствующему языку. Для реализации запросов к помеченным графам широко используются регулярные грамматики. Однако с их помощью бывает невозможно описать нужные запросы. Поэтому актуальна задача организации более выразительных запросов, используя КС-грамматики.

Для синтаксического анализа строки по произвольной КС-грамматике существуют различные алгоритмы. Например, Early parser [2] , CYK [8], GLR [6], GLL [4]. Алгоритм GLL имеет оптимальное время работы ($O(n^3)$ в худшем случае) и основан на идее нисходящего анализа, а значит более удобен для реализации. Поэтому для поиска пути используется именно этот алгоритм.

Таким образом, использование графов с метками на вершинах и рёбрах позволяет естественным образом представлять различные наборы данных, а обработка запросов необходима для эффективной работы с ними. КС-грамматики дают возможность писать выразительные за-

просы, при этом использование алгоритма GLL позволит быстро выполнять такие запросы.

1. Постановка задачи

Целью курсовой работы является реализация синтаксического анализа графов с помеченными вершинами и рёбрами. Для достижения этой цели поставлены перечисленные ниже задачи.

- В рамках проекта YaccConstructor [7] реализовать возможность поиска путей в графе с помеченными вершинами и рёбрами по заданной КС-грамматике.
- Реализовать удобный интерфейс для работы:
  - создание и выполнение запросов
  - получение и обработка результатов
- Провести апробацию и сравнить с существующими решениями.

2. Обзор

Для поиска путей в графе существует множество инструментов, позволяющих находить пути по регулярным грамматикам. Решений для поиска путей по КС-грамматике не так много, в особенности для графов с метками на вершинах и рёбрах.

В работе [5] решалась задача извлечения связного подграфа, состоящего из путей между двумя исходными вершинами, из графа с метками на вершинах и рёбрах. Класс подходящих путей описывается с помощью контекстно-свободной грамматики. Для синтаксического анализа используется алгоритм Earley, работающий в худшем случае за время $O(n^3)$. Однако, поиск путей производится не в исходном графе с метками на вершинах и рёбрах, а в преобразованном. Перед началом работы алгоритма из исходного получают новый двудольный граф с метками только на рёбрах. Новый граф имеет в 2 раза больше вершин и увеличивает число рёбер. Даже при небольших входных данных и для путей длины не больше 8 алгоритм работает 240 секунд, что делает его мало применимым на практике.

Одним из распространённых способов представлять данные в удобном для обработки виде является модель RDF. Данные, записанные в RDF, представляют собой набор триплетов субъект–предикат–объект. В совокупности они образуют помеченный ориентированный граф. Многие данные в биоинформатике представлены именно в таком формате.

Самым популярным языком для запросов к данным, представленным в формате RDF, является язык SPARQL [3]. Однако, он позволяет описывать только регулярные выражения. В статье [1] авторы описали алгоритм для поиска путей в RDF-графе, принадлежащих КС-языку, а также предложили язык csSPARQL, поддерживающий КС-грамматики. Показано, что сложность алгоритма $O((|N| * |G|)^3)$, где N — нетерминалы входной грамматики, G — RDF-граф.

3. Реализация прототипа синтаксического анализа графа

На кафедре Системного программирования в лаборатории языковых инструментов разрабатывается проект YaccConstructor. Это платформа для исследований в области синтаксического анализа, написанная на языке F#. YaccConstructor позволяет создавать синтаксические анализаторы и имеет модульную архитектуру. Для построения анализатора выбирается фронтенд для обработки грамматик, выполняются необходимые преобразования и по указанному генератору строится нужный результат.

В YaccConstructor есть абстрактная реализация алгоритма синтаксического анализа GLL. Исходная грамматика описывается на языке спецификации грамматик YARD. Затем генератором она преобразуется в файл на языке F#, содержащий необходимую для алгоритма информацию о грамматике.

Во время выполнения алгоритм перемещается по входному объекту в зависимости от текущей позиции в грамматике. Объект, в котором требуется найти пути, удовлетворяющие исходной КС-грамматике, должен реализовывать интерфейс IParserInput. Мною реализован этот интерфейс для графов с помеченными вершинами и рёбрами. Если текущая позиция — вершина, следующими позициями в графе являются все исходящие рёбра. Если текущая позиция на ребре, следующим является конечная вершина. Таким образом, алгоритмом проверяются все возможные пути в графе. Для тех путей, которые удовлетворяют запросу, прототип пока что возвращает только начальную и конечную позиции в графе.

Прототип опробован на графах, содержащих от 630 до 640 рёбер. Время работы в среднем составило 34 мс. Однако, для тестов использовались графы с небольшим разнообразием меток и несложной грамматикой. В дальнейшем будет произведена апробация на реальных данных.

4. Заключение

Результаты, достигнутые на данный момент:

- написан обзор предметной области
- реализован прототип, выполняющий поиск путей в графе с помеченными вершинами и рёбрами по заданной КС-грамматике

В дальнейшем планируется реализовать интерфейс для работы с запросами и обработки результатов, а также протестировать алгоритм на реальных данных и сравнить с существующими решениями.

Список литературы

[1] Context-free path queries on RDF graphs / Xiaowang Zhang, Zhiyong Feng, Xin Wang et al. // International Semantic Web Conference / Springer. — 2016. — P. 632–648.

[2] Earley Jay. An efficient context-free parsing algorithm // Communications of the ACM. — 1970. — Vol. 13, no. 2. — P. 94–102.

[3] Prud'hommeaux Eric, Seaborne Andy. SPARQL Query Language for RDF. W3C Recommendation, January 2008. — 2008.

[4] Scott Elizabeth, Johnstone Adrian. GLL parsing // Electronic Notes in Theoretical Computer Science. — 2010. — Vol. 253, no. 7. — P. 177–189.

[5] Sevon Petteri, Eronen Lauri. Subgraph queries by context-free grammars // Journal of Integrative Bioinformatics (JIB). — 2008. — Vol. 5, no. 2. — P. 157–172.

[6] Tomita Masaru. An efficient augmented-context-free parsing algorithm // Computational linguistics. — 1987. — Vol. 13, no. 1-2. — P. 31–46.

[7] YaccConstructor. YaccConstructor // YaccConstructor official page. — URL: `http://yaccconstructor.github.io`.

[8] Younger Daniel H. Recognition and parsing of context-free languages in time n3 // Information and control. — 1967. — Vol. 10, no. 2. — P. 189–208.