

Синтаксический анализ графов с использованием конъюнктивных грамматик

Азимов Р. Ш.,
rustam.azimov19021995@gmail.com,
Санкт-Петербургский государственный университет,
Лаборатория языковых инструментов JetBrains

11 февраля 2018 г.

Аннотация

Графы используются в качестве структуры данных во многих областях, например, биоинформатика, графовые базы данных. В этих областях часто необходимо вычислять некоторые запросы к большим графам. Ответом на такие запросы обычно является множество всех троек (A, m, n) , для которых существует путь в графе от вершины m до вершины n такой, что метки на ребрах этого пути образуют строку, выводимую из нетерминала данной контекстно-свободной грамматики A . Говорят, что такой тип запросов вычислен с использованием *реляционной семантики запросов*. Кроме того, существуют *конъюнктивные грамматики*, образующие более широкий класс грамматик, чем контекстно-свободные. Использование конъюнктивных грамматик в задаче синтаксического анализа графов позволит формулировать более сложные запросы к графу и решать более широкий круг задач. Известно, что задача вычисления запросов к графу с использованием реляционной семантики и конъюнктивных грамматик — неразрешима. В данной работе будет предложен алгоритм, вычисляющий приближенное решение данной задачи, а именно аппроксимацию сверху множества троек (A, m, n) . Предложенный алгоритм основан на матричных операциях, что позволяет повысить производительность, используя вычисления на графическом процессоре.

Ключевые слова: синтаксический анализ графов, конъюнктивные грамматики, транзитивное замыкание, матричные операции, вычисления на GPU

1 Введение

Графы используются в качестве структуры данных во многих областях, например, биоинформатика [12], графовые базы данных [9]. В этих областях часто необходимо вычислять некоторые запросы к большим графам. Одними из наиболее распространенных запросов к графам являются навигационные запросы. Результатом вычисления таких запросов является множество неявных отношений между вершинами графа, то есть путей в графе. Естественно выделять такие отношения — пометив ребра графа

символами из некоторого конечного алфавита и выделив необходимые пути в графе с помощью формальных грамматик (регулярные выражения, контекстно-свободные грамматики) над тем же алфавитом. Наиболее популярны запросы, использующие контекстно-свободные грамматики, так как КС-языки обладают большей выразительной мощностью, чем регулярные.

Также существуют *конъюнктивные грамматики* [11], образующие более широкий класс грамматик, чем контекстно-свободные. Использование конъюнктивных грамматик в задаче синтаксического анализа графов позволит формулировать более сложные запросы к графу и решать более широкий круг задач. Известно, что задача вычисления запросов к графу с использованием реляционной семантики и конъюнктивных грамматик — неразрешима [7]. Один из распространенных способов найти приближенное решение неразрешимой задачи — найти аппроксимацию решения (сверху или снизу).

В данной работе будет предложен алгоритм, вычисляющий приближенное решение задачи синтаксического анализа графов с использованием реляционной семантики запросов и конъюнктивных грамматик, а именно аппроксимацию сверху множества троек (A, m, n) . Предложенный алгоритм основан на матричных операциях, что позволяет повысить производительность, используя для вычислений графический процессор.

2 Обзор

В этом разделе мы определим задачу синтаксического анализа графов и обсудим основные подходы, применяемые для ее решения.

Пусть Σ — конечное множество терминальных символов. *Помеченным графом* будем называть пару $D = (V, E)$, где V является множеством вершин, а $E \subseteq V \times \Sigma \times V$ — множеством ребер с метками из алфавита Σ . Для пути π в графе D мы будем использовать $l(\pi)$ для обозначения слова, полученного конкатенацией меток на ребрах данного пути. Кроме того, мы будем писать $m\pi n$, чтобы указать, что существует путь из вершины $m \in V$ в вершину $n \in V$.

Результатом работы алгоритма синтаксического анализа графов с использованием формальной грамматики G обычно является множество всех троек (A, m, n) , для которых $m\pi n$ такой, что строка $l(\pi)$ выводима из нетерминала A грамматики G . Говорят, что такой тип запросов вычислен с использованием *реляционной семантики запросов* [7].

Традиционно использовали регулярные выражения в качестве грамматики G . Но в последнее время стало популярным использовать КС-грамматики, так как некоторые полезные запросы не могут быть описаны с помощью регулярных грамматик. Примером таких запросов являются классические запросы поиска всех вершин в графе, находящихся на одном уровне иерархии [1]. Рассмотренные алгоритмы синтаксического анализа графов принимают на вход КС-грамматики в *нормальной форме Хомского* [4].

Существует ряд алгоритмов синтаксического анализа графов с использованием реляционной семантики запросов и КС-грамматик [5; 7; 13], которые основаны на методе динамического программирования. Данные алгоритмы обобщают такие алгоритмы синтаксического анализа, как СΥΚ [8; 15] и Earley [6]. В работе [7] для заданного графа $D = (V, E)$ и КС-грамматики

$G = (N, \Sigma, P)$, определяются *контекстно-свободные отношения* $R_A \subseteq V \times V$ для каждого $A \in N$ следующим образом:

$$R_A = \{(n, m) \mid \exists \pi m \ (l(\pi) \in L(G_A))\},$$

где $L(G_A)$ — язык, порожденный грамматикой G со стартовым нетерминалом A .

Вся работа алгоритма [7] сводится к вычислению контекстно-свободных отношений R_A для каждого $A \in N$. Кроме того, существует алгоритм синтаксического анализа графов с использованием реляционной семантики запросов и КС-грамматик, вычисляющий данные контекстно-свободные отношения R_A используя матричное транзитивное замыкание [2]. Данный алгоритм обобщает алгоритм Вэлианта [14] и сводится к ряду умножений Булевых матриц.

Также существуют *конъюнктивные грамматики* [11], образующие более широкий класс грамматик, чем контекстно-свободные. Как и в случае КС-грамматик мы рассматриваем только конъюнктивные грамматики в бинарной нормальной форме [10]. Мы не выделяем стартовый нетерминал, так как его можно будет определить во время синтаксического анализа графа. Так как для каждой конъюнктивной грамматики можно построить эквивалентную ей грамматику в бинарной нормальной форме, то достаточно рассмотреть только грамматики следующего вида.

Конъюнктивная грамматика — это тройка $G = (N, \Sigma, P)$, где N — конечное множество нетерминальных символов, Σ — конечное множество терминальных символов и P — конечное множество правил следующего типа:

- $A \rightarrow B_1 C_1 \& \dots \& B_m C_m$, for $m \geq 1$, $A, B_i, C_i \in N$,
- $A \rightarrow x$, for $A \in N$ and $x \in \Sigma$.

Мы будем использовать запись $A \xrightarrow{*} w$, чтобы указать, что строка $w \in \Sigma^*$ может быть получена из нетерминала A некоторой последовательностью применений правил конъюнктивной грамматики, где отношение \rightarrow определено следующим образом:

- При применении правила $A \rightarrow B_1 C_1 \& \dots \& B_m C_m \in P$, любой подтерм A любого терма может быть перезаписан подтермом $(B_1 C_1 \& \dots \& B_m C_m)$:

$$\dots A \dots \rightarrow \dots (B_1 C_1 \& \dots \& B_m C_m) \dots$$

- Конъюнкция нескольких одинаковых строк из Σ^* может быть перезаписана одной такой строкой: для любого $w \in \Sigma^*$,

$$\dots (w \& \dots \& w) \dots \rightarrow \dots w \dots$$

Языком, сгенерированным конъюнктивной грамматикой $G = (N, \Sigma, P)$ со стартовым нетерминалом $S \in N$, будем называть

$$L(G_S) = \{w \in \Sigma^* \mid S \xrightarrow{*} w\}.$$

Определение контекстно-свободных отношений R_A естественным образом обобщается до определения конъюнктивных отношений R_A заменой

КС-грамматики G на конъюнктивную. Таким образом, задача синтаксического анализа графов с использованием реляционной семантики запросов и конъюнктивных грамматик сводится к поиску конъюнктивных отношений R_A для всех нетерминалов A . Но также известен факт, что данная задача является неразрешимой [7]. Поэтому единственный известный алгоритм для решения данной задачи, описанный в [16], дает приближенный результат. Данный алгоритм используется для статического анализа программ и аппроксимирует сверху конъюнктивные отношения R_A , то есть точное решение является подмножеством предложенного решения.

3 Существующие работы

Ряд алгоритмов синтаксического анализа графов с использованием реляционной семантики запросов и КС-грамматик [5; 7; 13] демонстрируют низкую производительность на больших графах. Одной из самых популярных техник, используемых для увеличения производительности при работе с большими объемами данных, является использование графического процессора для вычислений, но перечисленные алгоритмы не позволяют эффективно применить данную технику.

В то время как алгоритм синтаксического анализа графов с использованием реляционной семантики запросов и КС-грамматик, вычисляющий матричное транзитивное замыкание [2] активно использует матричные операции и позволяет эффективно использовать вычисления на графическом процессоре [3].

Но все перечисленные алгоритмы работают только с КС-грамматиками и не позволяют обрабатывать запросы, описанные с помощью конъюнктивных грамматик. В работе [16] описан единственный известный алгоритм, работающий с конъюнктивными грамматиками. Но данный алгоритм принимает на вход только определенный подкласс конъюнктивных грамматик, а именно линейные конъюнктивные грамматики.

Таким образом, целью данной работы является создание первого алгоритма синтаксического анализа графов с использованием реляционной семантики запросов, который работает с произвольной конъюнктивной грамматикой. Кроме того, данный алгоритм должен активно использовать матричные операции, что позволит эффективно использовать графический процессор для вычислений.

4 Алгоритм

5 Эксперименты

6 Заключение

Список литературы

1. *Abiteboul S., Hull R., Vianu V.* Foundations of databases: the logical level. — Addison-Wesley Longman Publishing Co., Inc., 1995.

2. *Azimov R., Grigorev S.* Context-Free Path Querying by Matrix Multiplication. — 2018.
3. *Che S., Beckmann B. M., Reinhardt S. K.* Programming GPGPU Graph Applications with Linear Algebra Building Blocks // International Journal of Parallel Programming. — 2016. — C. 1–23.
4. *Chomsky N.* On certain formal properties of grammars // Information and control. — 1959. — T. 2, № 2. — C. 137–167.
5. Context-free path queries on RDF graphs / X. Zhang [и др.] // International Semantic Web Conference. — Springer. 2016. — C. 632–648.
6. *Grune D., Jacobs C. J. H.* Parsing Techniques (Monographs in Computer Science). — Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006. — ISBN 038720248X.
7. *Hellings J.* Conjunctive context-free path queries. — 2014.
8. *Kasami T.* AN EFFICIENT RECOGNITION AND SYNTAXANALYSIS ALGORITHM FOR CONTEXT-FREE LANGUAGES.Тех. отч. / DTIC Document. — 1965.
9. *Mendelzon A., Wood P.* Finding Regular Simple Paths in Graph Databases // SIAM J. Computing. — 1995. — T. 24, № 6. — C. 1235–1258.
10. *Okhotin A.* Conjunctive and Boolean grammars: the true general case of the context-free grammars // Computer Science Review. — 2013. — T. 9. — C. 27–59.
11. *Okhotin A.* Conjunctive grammars // Journal of Automata, Languages and Combinatorics. — 2001. — T. 6, № 4. — C. 519–535.
12. Quantifying variances in comparative RNA secondary structure prediction / J. W. Anderson [и др.] // BMC bioinformatics. — 2013. — T. 14, № 1. — C. 149.
13. *Sevon P., Eronen L.* Subgraph queries by context-free grammars // Journal of Integrative Bioinformatics. — 2008. — T. 5, № 2. — C. 100.
14. *Valiant L. G.* General context-free recognition in less than cubic time // Journal of computer and system sciences. — 1975. — T. 10, № 2. — C. 308–315.
15. *Younger D. H.* Recognition and parsing of context-free languages in time n^3 // Information and control. — 1967. — T. 10, № 2. — C. 189–208.
16. *Zhang Q., Su Z.* Context-sensitive data-dependence analysis via linear conjunctive language reachability // Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages. — ACM. 2017. — C. 344–358.