# Синтаксический анализ графов с использованием конъюнктивных грамматик

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

4 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. 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 над тем же алфавитом. Наиболее популярны запросы, использующие контекстно-свободные грамматики. 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 [7], графовые базы данных [5]. В этих областях часто необходимо вычислять некоторые запросы к большим графам. Один из наиболее распространенных запросам 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.

Ответом на запросы с использованием КС-грамматик 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, выводимою из нетерминала $A$ данной КС-грамматики. Говорят, что такой тип запросов вычислен с использованием *реляционной семантики запросов* [4].

Существует ряд алгоритмов синтаксического анализа графов с использованием реляционной семантики запросов и КС-грамматик [3; 4; 8]. Данные алгоритмы 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.

Кроме того, существует алгоритм синтаксического анализа графов с использование реляционной семантики запросов и КС-грамматик, вычисляющий матричное транзитивное замыкание. Активное использование матричных операций в данном алгоритме позволяет эффективно использовать вычисления на графическом процессоре [1].

Также существуют *конъюнктивные грамматики* [6], 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 [4]. 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 вычисления на графическом процессоре.

# 2 Обзор

# 3 Существующие работы

# 4 Определения

В этом разделе мы дадим несколько определений, связанных с задачей синтаксического анализа графов.

Пусть $\Sigma$ — 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 $\Sigma$. Для пути $\pi$ в графе $D$ мы будем использовать $l(\pi)$ для обозначения слова, полученного конкатенацией меток на ребрах данного пути. Кроме того, мы будем писать $n\pi m$, чтобы указать, что существует путь из вершины $n \in V$ в вершину $m \in V$.

Как и в работе [4], мы рассматриваем только КС-грамматики в *нормальной форме Хомского* [2]. Мы не выделяем стартовый нетерминал, так как его можно будет определить во время синтаксического анализа графа. Так как для каждой КС-грамматики можно построить эквивалентную ей грамматику в нормальной форме Хомского, то достаточно рассмотреть только грамматики следующего вида. *Контестно-свободная грамматика* — это тройка $G = (N, \Sigma, P)$, где $N$ — конечное множество нетерминальных символов, $\Sigma$ — конечное множество терминальных символов и $P$ — конечное множество правил следующего типа:

- $A \to BC$, для $A, B, C \in N$,
- $A \to x$, для $A \in N$, $x \in \Sigma$.

Заметим, что мы не рассматриваем правила вида $A \to \varepsilon$, где $\varepsilon$ обозначает пустую строку. Это не ограничивает применимость нашего алгоритма, так как только пустые пути вида $m\pi m$ соответсвуют пустой строке $\varepsilon$.

Мы будем использовать запись $A \xrightarrow{*} w$, чтобы указать, что строка $w \in \Sigma^*$ может быть получена из нетерминала $A$ некоторой последовательностью применений правил грамматики. *Языком*, сгенерированным грамматикой $G = (N, \Sigma, P)$ со стартовым нетерминалом $S \in N$, будем называть

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

Для заданного графа $D = (V, E)$ и КС-грамматики $G = (N, \Sigma, P)$, определим *контекстно-свободные отношения* $R_A \subseteq V \times V$, для каждого $A \in N$ следующим образом:

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

Также определим бинарную операцию $(\ \cdot\ )$ на произвольных подмножествах $N_1, N_2$ множества нетерминальных символов $N$ грамматики $G = (N, \Sigma, P)$ следующим образом:

$$N_1 \cdot N_2 = \{A \mid \exists B \in N_1, \exists C \in N_2 \text{ such that } (A \to BC) \in P\}.$$

Используя операцию ( $\cdot$ ) в качестве операции умножения подмножеств множества $N$ и объединение в качестве сложения, мы можем определить *матричное умножение*, $a \times b = c$, где $a$ и $b$ — матрицы подходящего размера, элементы которых являются подмножествами множества $N$, следующим образом:

$$c_{i,j} = \bigcup_{k=1}^{n} a_{i,k} \cdot b_{k,j}.$$

Также мы определим *матричное транзитивное замыкание* квадратной матрицы $a$, как $a^{cf} = a^{(1)} \cup a^{(2)} \cup \cdots$, где $a^{(1)} = a$ и

$$a^{(i)} = a^{(i-1)} \cup (a^{(i-1)} \times a^{(i-1)}),\ i \geq 2.$$

# 5 Сведение синтаксического анализа графов к поиску транзитивного замыкания

# 6 Алгоритм

# 7 Апробация

# 8 Заключение

# Список литературы


1. *Che S.*, *Beckmann B. M.*, *Reinhardt S. K.* Programming GPGPU Graph Applications with Linear Algebra Building Blocks // International Journal of Parallel Programming. — 2016. — С. 1—23.
2. *Chomsky N.* On certain formal properties of grammars // Information and control. — 1959. — Т. 2, № 2. — С. 137—167.
3. Context-free path queries on RDF graphs / X. Zhang [и др.] // International Semantic Web Conference. — Springer. 2016. — С. 632—648.
4. *Hellings J.* Conjunctive context-free path queries. — 2014.
5. *Mendelzon A.*, *Wood P.* Finding Regular Simple Paths in Graph Databases // SIAM J. Computing. — 1995. — Т. 24, № 6. — С. 1235—1258.
6. *Okhotin A.* Conjunctive grammars // Journal of Automata, Languages and Combinatorics. — 2001. — Т. 6, № 4. — С. 519—535.
7. Quantifying variances in comparative RNA secondary structure prediction / J. W. Anderson [и др.] // BMC bioinformatics. — 2013. — Т. 14, № 1. — С. 149.
8. *Sevon P.*, *Eronen L.* Subgraph queries by context-free grammars // Journal of Integrative Bioinformatics. — 2008. — Т. 5, № 2. — С. 100.