

Приложения теории формальных языков и синтаксического анализа

Семён Григорьев

29 августа 2019 г.

Содержание

1	Введение	3
2	Постановка задачи	3
3	О разрешимости задачи поиска путей с ограничениями в терминах языков	4
4	Алгоритм на матричных операциях	4
4.1	Транзитивное замыкание графа	4
4.2	Нормальная форма Хомского	4
4.3	Алгоритм СУК	4
4.4	Алгоритм для графов на основе СУК	4
4.5	Алгоритм на основе матриц	4
4.6	Конъюнктивные и булевы грамматики	4
4.6.1	Определения	4
4.6.2	Для графов	5
4.7	Особенности реализации матричного алгоритма	5
4.8	Обзор	5
4.9	Вопросы и задачи	5
5	Через тензорное произведение	5
5.1	Рекурсивные автоматы и сети	6
5.2	Тензорное произведение	6
5.3	Алгоритм	8
5.4	Примеры	9
5.5	Вопросы и задачи	28
6	Сжатое представление леса разбора	29
6.1	Дерево вывода	29
6.2	Неоднозначные грамматики	29
6.3	Лес разбора как представление контекстно-свободной грамматики	29
6.4	Вопросы и задачи	29
7	Алгоритм на основе восходящего анализа	30
7.1	Восходящий синтаксический анализ	30
7.2	Вопросы и задачи	30
8	Алгоритм на основе нисходящего анализа	30
8.1	Нисходящий синтаксический анализ	31

9	От CFPQ к вычислению Datalog-запросов	31
9.1	Datalog	31
9.2	КС-запрос как запрос на Datalog	31
9.3	Обобщение GLL для вычисления Datalog-запросов	31

1 Введение

Одна из классических задач, связанных с анализом графов — это поиск путей в графе. Возможны различные формулировки этой задачи. В некоторых случаях необходимо выяснить, существует ли путь с определёнными свойствами между двумя выбранными вершинами. В других же ситуациях необходимо найти все пути в графе, удовлетворяющие некоторым свойствам.

Так или иначе, на практике часто требуется указать, что интересующие нас пути должны обладать каким-либо специальными свойствами. Иными словами, наложить на пути некоторые ограничения. Например, указать, что искомый путь должен быть простым, кратчайшим, гамильтоновым и так далее.

Один из способов задавать ограничения на пути в графе основан на использовании формальных языков. Базовое определение языка говорит нам, что язык — это множество слов над некоторым алфавитом. Если рассмотреть граф, рёбра которого помечены символами из алфавита, то путь в графе будет задавать слово: достаточно соединить последовательно символы, лежащие на рёбрах пути. Множество же таких путей будет задавать множество слов или язык. Таким образом, если мы хотим найти некоторое множество путей в графе, то в качестве ограничения можно описать язык, который должно задавать это множество. Иными словами, задача поиска путей может быть сформулирована следующим образом: необходимо найти такие пути в графе, что слова, получаемые конкатенацией меток их рёбер, принадлежат заданному языку. Такой класс задач будем называть задачами поиска путей с ограничениям в терминах формальных языков.

Рассмотрим различные варианты постановки задачи.

Различные алгоритмы решения.

2 Постановка задачи

Пусть $\mathcal{G} = \langle V, E, L \rangle$ — конечный ориентированный граф с метками на рёбрах, где V — конечное множество вершин, E — конечное множество рёбер, L — конечное множество или алфавит меток. Рёбра будем представлять в виде упорядоченных троек из $V \times L \times V$. Путём π в графе \mathcal{G} будем называть последовательность рёбер такую, что для любых двух последовательных рёбер $e_1 = (u_1, l_1, v_1)$ и $e_2 = (u_2, l_2, v_2)$ в этой последовательности, конечная вершина первого ребра является начальной вершиной второго, то есть $v_1 = u_2$. Будем обозначать путь из вершины v_0 в вершину v_n как $v_0\pi v_n = e_0, e_1, \dots, e_{n-1} = (v_0, l_0, v_1), (v_1, l_1, v_2), \dots, (v_{n-1}, l_n, v_n)$.

Функция $\omega(\pi) = \omega((v_0, l_0, v_1), (v_1, l_1, v_2), \dots, (v_{n-1}, l_n, v_n)) = l_0 \cdot l_1 \cdot \dots \cdot l_n$ строит слово по пути посредством конкатенации меток рёбер вдоль этого пути. Очевидно, для пустого пути данная функция будет возвращать пустое слово, а для пути длины $n > 0$ — непустое слово длины n .

Пусть $G = \langle \Sigma, N, P \rangle$ — контекстно-свободная грамматика. Будем считать, что $L \subseteq \Sigma$. Мы не фиксируем стартовый нетерминал в определении грамматики, поэтому, чтобы описать язык, задаваемый ей, нам необходимо отдельно зафиксировать стартовый нетерминал. Таким образом, будем говорить, что $L(G, N_i) = \{w | N_i \xrightarrow{*}_G w\}$ — это язык задаваемый грамматикой G со стартовым нетерминалом N_i .

Задача достижимости:

Задача поиска путей:

3 О разрешимости задачи поиска путей с ограничениями в терминах языков

Сведение к задаче о пересечении с регулярным.

Замкнутость регулярных.

Проверка пустоты.

Замкнутость контекстно-свободных. Проверка пустоты.

Про другие классы языков: конъюнктивные, булевы, многокомпонентные.

4 Алгоритм на матричных операциях

4.1 Транзитивное замыкание графа

Флойд-Уоршал, матрицы.

Рассуждения про субкубичность. Про то, что булево полукольцо.

4.2 Нормальная форма Хомского

Данный алгоритм накладывает ограничение на форму грамматики: грамматика должна быть в “ослабленной” нормальной форме Хомского.

Определение НФХ.

Любую КС грамматику можно преобразовать в НФХ.

4.3 Алгоритм СҮК

Построение и заполнение таблицы.

4.4 Алгоритм для графов на основе СҮК

Обобщение. Смотрим на транзитивное замыкание.

4.5 Алгоритм на основе матриц

Ссылка на Валианта [?].

Оригинальные матрицы (Рустам) [1]

Кратчайшие.

Одно дерево?

4.6 Конъюнктивные и булевы грамматики

4.6.1 Определения

Охотин [?].

4.6.2 Для графов

Классическая семантика — работает для конъюнктивных для любых графов. Для Булевых и конъюнктивных только для графов без циклов.

Альтернативная семантика (DataLog). Трактую конъюнкцию как в даталоге. Тогда всё хорошо. Это похоже на даталог через линейную алгебру [?]

4.7 Особенности реализации матричного алгоритма

Кое-что про наши реализации [6]

Разреженные матрицы, плотные матрицы, GraphBLAS¹, GPGPU, CUTLASS².

Расположение в памяти: хорошо, когда всё влезло на одну карту.

Распределённые решения. Через GraphBLAS

4.8 Обзор

Китайцы [?], Брэдфорд [?], Ли [?], Хеллингс [?], OpenCypher [4].

Рассуждения про асимптотику.

Субкубический для частного случая (Брэдфорд) [2].

4.9 Вопросы и задачи

1. !!!

5 Через тензорное произведение

Предыдущий подход позволяет выразить задачу поиска путей с ограничениями в терминах нормальных языков через набор матричных операций. Это позволяет использовать высокопроизводительные библиотеки и массовопараллельные архитектуры и вообще круто. Однако, такой подход требует, чтобы грамматика находилась в ослабленной нормальной форме Хомского, что приводит к её разрастанию. Можно ли как-то избежать этого?

В данном разделе мы попробуем предложить альтернативное сведение задачи поиска путей к матричным операциям. В результате мы сможем избежать преобразования грамматики в ОНФХ, однако, матрицы, с которыми нам придётся работать, будут существенно большего размера.

В основе подхода лежит использование рекурсивных сетей или рекурсивных автоматов в качестве представления контекстно-свободных грамматик.

¹!!!

²Репозиторий библиотеки CUTLASS: <https://github.com/NVIDIA/cutlass>

5.1 Рекурсивные автоматы и сети

Рекурсивный автомат или сеть — это представление контекстно-свободных грамматик, обобщающее конечные автоматы. В нашей работе мы будем придерживаться термина **рекурсивный автомат**. Классическое определение рекурсивного автомата выглядит следующим образом.

Определение 5.1. Рекурсивный автомат — это

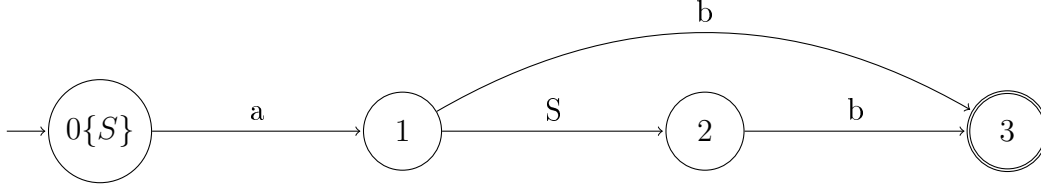


Рис. 1: Рекурсивный автомат для грамматики!!!

5.2 Тензорное произведение

Тензорное произведение матриц или произведение Кронекера — это бинарная операция, обозначаемая \otimes и определяемая следующим образом.

Определение 5.2. Пусть даны две матрицы: A размера $m \times n$ и B размера $p \times q$. Произведение Кронекера или тензорное произведение матриц A и B — это блочная матрица C размера $mp \times nq$, вычисляемая следующим образом:

$$C = A \otimes B = \begin{pmatrix} A_{0,0}B & \cdots & A_{0,n-1}B \\ \vdots & \ddots & \vdots \\ A_{m-1,0}B & \cdots & A_{m-1,n-1}B \end{pmatrix}$$

Пример 5.1.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} & 2 \begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \\ 3 \begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} & 4 \begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 7 & 8 & 10 & 12 & 14 & 16 \\ 9 & 10 & 11 & 12 & 18 & 20 & 22 & 24 \\ 13 & 14 & 15 & 16 & 26 & 28 & 30 & 32 \\ 15 & 18 & 21 & 24 & 20 & 24 & 28 & 32 \\ 27 & 30 & 33 & 36 & 36 & 40 & 44 & 48 \\ 39 & 42 & 45 & 48 & 52 & 56 & 60 & 64 \end{pmatrix} \quad (1)$$

Заметим, что для определения тензорного произведения матриц достаточно определить операцию умножения на элементах исходных матриц. Также отметим, что произведение Кронекера не является коммутативной. При этом всегда существуют такие две матрицы перестановок P и Q , что $A \otimes B = P(B \otimes A)Q$. Это свойство потребуется нам в дальнейшем.

Теперь перейдём к графам. Сперва дадим классическое определение тензорного произведения двух неориентированных графов.

Определение 5.3. Пусть даны два графа: $\mathcal{G}_1 = \langle V_1, E_1 \rangle$ и $\mathcal{G}_2 = \langle V_2, E_2 \rangle$. Тензорным произведением этих графов будем называть граф $\mathcal{G}_3 = \langle V_3, E_3 \rangle$, где $V_3 = V_1 \times V_2$, $E_3 = \{((v_1, v_2), (u_1, u_2)) \mid (v_1, u_1) \in E_1 \text{ и } (v_2, u_2) \in E_2\}$.

Иными словами, тензорным произведением двух графов является граф, такой что:

1. множество вершин — это прямое произведение множеств вершин исходных графов;
2. ребро между вершинами $v = (v_1, v_2)$ и $u = (u_1, u_2)$ существует тогда и только тогда, когда существуют рёбра между парами вершин v_1, u_1 и v_2, u_2 в соответствующих графах.

Для того, чтобы построить тензорное произведение ориентированных графов, необходимо в предыдущем определении, в условии существования ребра в результирующем графе, дополнительно потребовать, чтобы направления рёбер совпадали. Данное требование получается естественным образом, если считать, что пары вершин, задающие ребро упорядочены, поэтому формальное определение отличаться не будет.

Осталось добавить метки к рёбрам. Это приведёт к логичному усилению требования к существованию ребра: метки рёбер в исходных графах должны совпадать. Таким образом, мы получаем следующее определение тензорного произведения ориентированных графов с метками на рёбрах.

Определение 5.4. Пусть даны два ориентированных графа с метками на рёбрах: $\mathcal{G}_1 = \langle V_1, E_1, L_1 \rangle$ и $\mathcal{G}_2 = \langle V_2, E_2, L_2 \rangle$. Тензорным произведением этих графов будем называть граф $\mathcal{G}_3 = \langle V_3, E_3, L_3 \rangle$, где $V_3 = V_1 \times V_2$, $E_3 = \{((v_1, v_2), l, (u_1, u_2)) \mid (v_1, l, u_1) \in E_1 \text{ и } (v_2, l, u_2) \in E_2\}$, $L_3 = L_1 \cap L_2$.

Нетрудно заметить, что матрица смежности графа \mathcal{G}_3 равна тензорному произведению матриц смежностей исходных графов \mathcal{G}_1 и \mathcal{G}_2 .

Рассмотрим пример. В качестве одного из графов возьмём рекурсивный автомат, построенный ранее!!!. Его матрица смежности выглядит следующим образом.

$$M_1 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

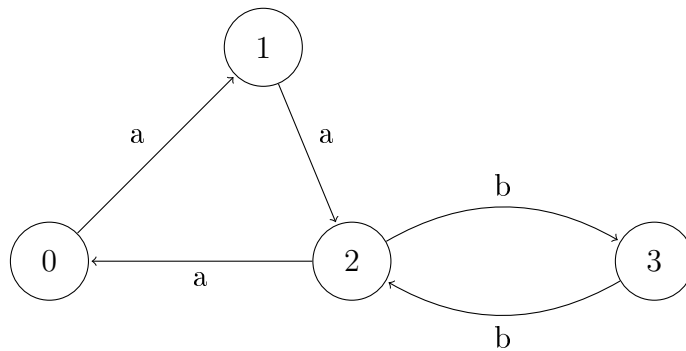


Рис. 2: The input graph

Второй граф представлен на изображении 2. Его матрица смежности имеет следующий вид.

$$M_2 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [a] & \cdot \\ [a] & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

Вычислим $M_1 \otimes M_2$.

$$M_3 = M_1 \otimes M_2 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \otimes \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [a] & \cdot \\ [a] & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix} =$$

$$= \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad (2)$$

5.3 Алгоритм

Идея алгоритма основана на обобщении пересечения двух конечных автоматов до пересечения рекурсивной сети, построенной по грамматике, со входным графом.

Несколько наблюдений Путь из нетерминалов и терминалов. При этом должны быть пути из терминалов. Иначе не задать язык. Будем насыщать граф рёбрами, помеченными нетерминалами.

По грамматике строим автомат. г 13

В цикле: пересекли через тензорное произведение, замкнули через обычное матричное произведение, чтобы найти пути из начальной в конечную в грамматике, поставили в соответствующие ячейки нетерминалы, добавили соответствующие рёбра в исходный граф.

Можно вычислять только разницу. Для этого, правда, потребуется держать ещё одну матрицу. И надо проверять, что вычислительно дешевле: поддерживать разницу и потом каждый раз поэлементно складывать две матрицы или каждый раз вычислять полностью произведение.

Всего несколько матриц. Разреженные. Необходимо отметить, что для реальных графов и запросов результат тензорного произведения будет очень разрежен. На готовых либах должно быть быстро.

5.4 Примеры

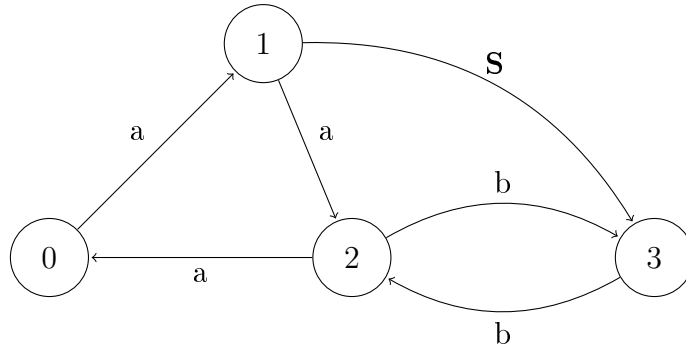
Рассмотрим ряд примеров работы описанного алгоритма. Разница выделена жирным.

Пример 1.

Худший случай. Такой же как и для матричного.

$$tc(M_3) = \left(\begin{array}{cccc|cccc|cccc|cccc} \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right) \quad (3)$$

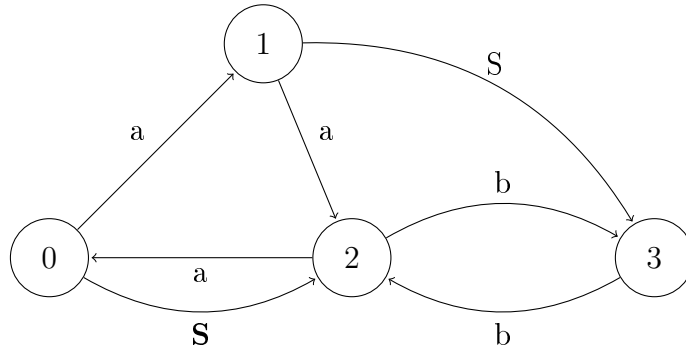
Мы видим, что в результате транзитивного замыкания появилось новое ребро с меткой ab из вершины $(0, 1)$ в вершину $(3, 3)$. Так как вершина 0 является стартовой в рекурсивном автомате, а 3 является финальной, то слово вдоль пути из вершины 1 в вершину 3 во входном графе выводимо из нетерминала S . Это означает, что в графе должно быть добавлено ребро из 0 в 3 с меткой S , после чего граф будет выглядеть следующим образом:



Матрица смежности обновлённого графа:

$$M_2 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [a] & [S] \\ [a] & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

Итерация закончена. Возвращаемся к началу цикла и вновь вычисляем тензорное произведение.



И матрица смежности:

$$M_2 = \begin{pmatrix} \cdot & [a] & [S] & \cdot \\ \cdot & \cdot & [a] & [S] \\ [a] & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

Следующая итерация основного цикла.

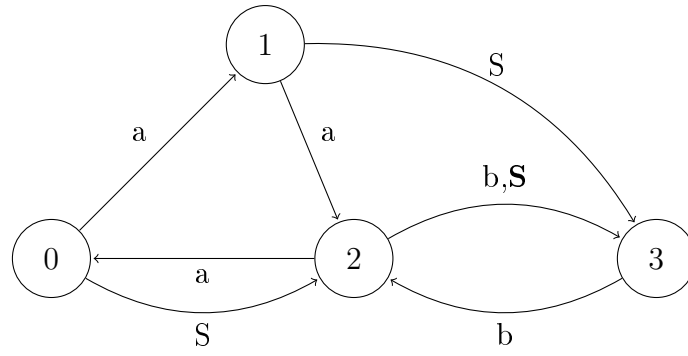
$$M_3 = M_1 \otimes M_2 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \otimes \begin{pmatrix} \cdot & [a] & [S] & \cdot \\ \cdot & \cdot & [a] & [S] \\ [a] & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix} =$$

$$= \left(\begin{array}{cccc|cccc|cccc|cccc} \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right) \quad (6)$$

Транзитивное замыкание:

$$tc(M_3) = \left(\begin{array}{c|c|c|c} \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [a] & \cdot \\ [a] & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & [aS] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [aS] & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & [aSb] & \cdot \\ \cdot & \cdot & \cdot & [ab] \\ \cdot & \cdot & \cdot & [aSb] \\ \cdot & \cdot & \cdot & \cdot \end{matrix} \\ \hline \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & [S] & \cdot \\ \cdot & \cdot & \cdot & [S] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & [Sb] \\ \cdot & \cdot & [Sb] & \cdot \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{matrix} \\ \hline \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & [b] & \cdot \end{matrix} \\ \hline \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix} \end{array} \right) \quad (7)$$

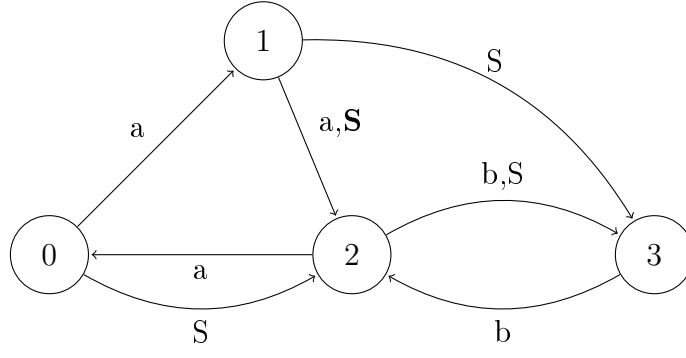
Обновлённый граф:



И матрица смежности:

$$M_2 = \begin{pmatrix} \cdot & [a] & [S] & \cdot \\ \cdot & \cdot & [a] & [S] \\ [a] & \cdot & \cdot & [b, \mathbf{S}] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

Следующая итерация основного цикла.



И матрица смежности:

$$M_2 = \begin{pmatrix} \cdot & [a] & [S] & \cdot \\ \cdot & \cdot & [a, S] & [S] \\ [a] & \cdot & \cdot & [b, S] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

Следующая итерация основного цикла.

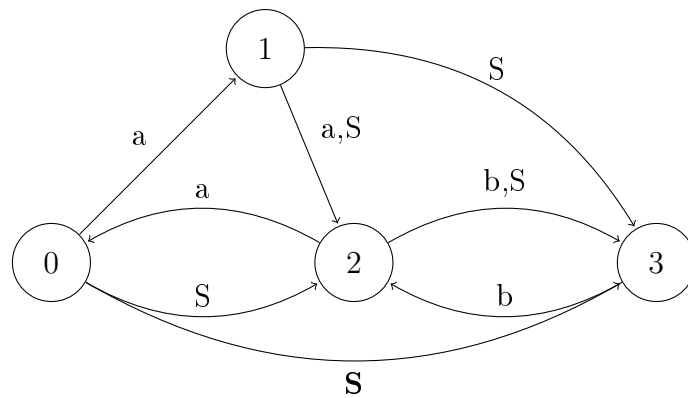
$$M_3 = M_1 \otimes M_2 = \begin{pmatrix} \cdot & [a] & \cdot & \cdot \\ \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & [b] \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \otimes \begin{pmatrix} \cdot & [a] & [S] & \cdot \\ \cdot & \cdot & [a, S] & [S] \\ [a] & \cdot & \cdot & [b, S] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix} =$$

$$= \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad (10)$$

Транзитивное замыкание:

$$tc(M_3) = \left(\begin{array}{cccc|cccc|cccc|cccc} \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & [aS] & [aS] & \cdot & \cdot & [aSb] & [\mathbf{aSb}] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & [aS] & [aS] & \cdot & \cdot & [aSb] \\ \cdot & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & [aS] & \cdot & \cdot & \cdot & \cdot & [aSb] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & [Sb] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & [S] & \cdot & \cdot & [Sb] & [\mathbf{Sb}] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & [Sb] & [b] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right) \quad (11)$$

Обновлённый граф:



И матрица смежности:

$$M_2 = \begin{pmatrix} \cdot & [a] & [S] & [\mathbf{S}] \\ \cdot & \cdot & [a, S] & [S] \\ [a] & \cdot & \cdot & [b, S] \\ \cdot & \cdot & [b] & \cdot \end{pmatrix}$$

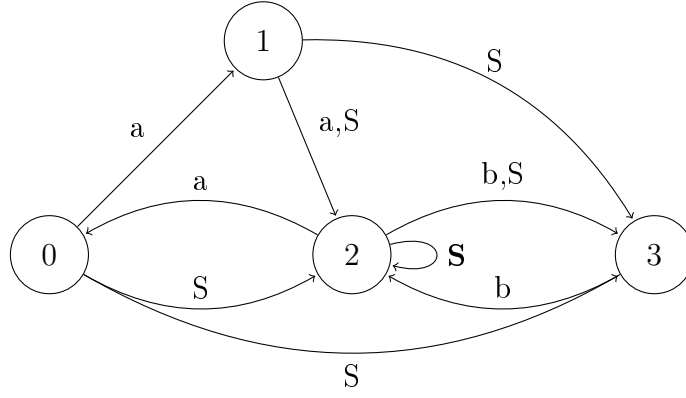
И наконец последняя содержательная итерация основного цикла.

[illegible]

Транзитивное замыкание:

[illegible]

Обновлённый граф:



И матрица смежности:

$$M_2 = \begin{pmatrix} . & [a] & [S] & [S] \\ . & . & [a, S] & [S] \\ [a] & . & [S] & [b, S] \\ . & . & [b] & . \end{pmatrix}$$

Следующая итерация не приведёт к изменению графа. Читатель может убедиться в этом самостоятельно. Соответственно алгоритм можно завершать.

Пример 2.

В данном примере мы увидим, как структура грамматики и, соответственно, рекурсивной сети, влияет на процесс вычислений.

Интуитивно понятно, что чем меньше состояний в рекурсивной сети, тем лучше. То есть желательно получить как можно более компактное описание контекстно-свободного языка.

Для примера возьмём в качестве КС языка язык Дика на одном типе скобок и опишем его двумя различными грамматиками. Первая грамматика классическая:

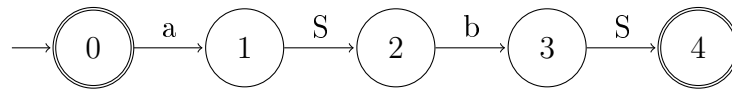
$$G_1 = \langle \{a, b\}, \{S\}, \{S \rightarrow a S b S \mid \varepsilon\} \rangle$$

Во второй грамматике мы будем использовать конструкции регулярных выражений в правой части правил. То есть вторая грамматика находится в EBNF [?].

$$G_2 = \langle \{a, b\}, \{S\}, \{S \rightarrow (a S b)^*\} \rangle$$

Построим рекурсивные автоматы N_1 и N_2 для этих грамматик.

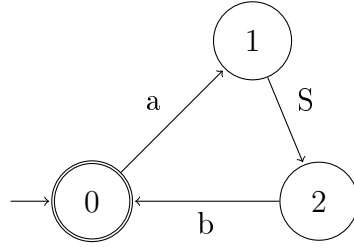
Рекурсивный автомат N_1 для грамматики G_1 :



Матрица смежности N_1 :

$$M_1^1 = \begin{pmatrix} . & [a] & . & . & . \\ . & . & [S] & . & . \\ . & . & . & [b] & . \\ . & . & . & . & [S] \\ . & . & . & . & . \end{pmatrix}$$

Рекурсивный автомат N_2 для грамматики G_2 :



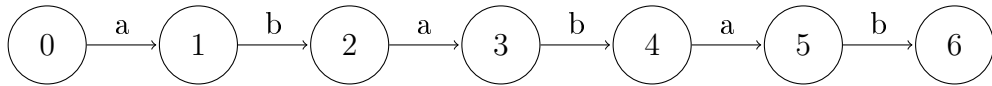
Матрица смежности N_2 :

$$M_1^2 = \begin{pmatrix} \cdot & [a] & \cdot \\ \cdot & \cdot & [S] \\ [b] & \cdot & \cdot \end{pmatrix}$$

Первое, очевидное, наблюдение — количество состояний в N_2 меньше, чем в N_1 . Это значит, что матрицы будут меньше, считать быстрее.

Для того, чтобы проще было сделать второе, сперва выполним пошагово алгоритм для двух заданных рекурсивных сетей.

Вход возьмём линейный:



Сразу дополним матрицу смежности нетерминалами, выводящими пустую строку, и получим следующую матрицу:

$$M_2 = \begin{pmatrix} [S] & [a] & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & [S] & [b] & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [S] & [a] & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & [S] & [b] & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [S] & [a] & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] \end{pmatrix}$$

Сперва запустим алгоритм на входе и N_2 . Первый шаг первой итерации — вычисление тензорного произведения $M_1^2 \otimes M_2$.

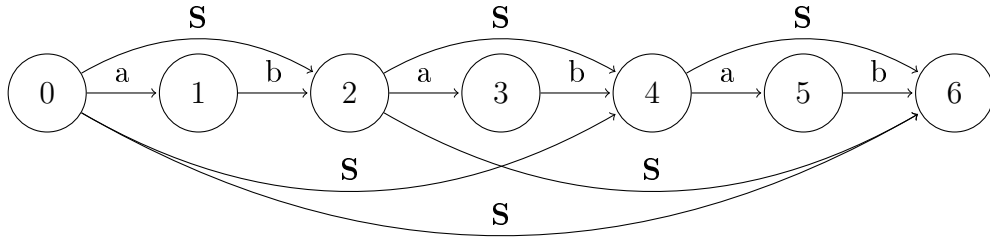
[illegible]

Опустим промежуточные шаги вычисления транзитивного замыкания M_3 и сразу пред-

ставим конечный результат:

$$tc(M_3) = \left(\begin{array}{ccc|ccc|ccc} \cdot & \cdot & [aSb] & \cdot & [aSbaSb] & \cdot & [aSbaSbaSb] & \cdot & [a] & \cdot & [aSba] & \cdot & [aSbaSba] & \cdot & \cdot & [aS] & \cdot & [aSbaS] & \cdot & [aSbaSbaS] & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [aSb] & \cdot & [aSbaSb] & \cdot & \cdot & \cdot & [a] & \cdot & [aSba] & \cdot & \cdot & \cdot & [aS] & \cdot & [aSbaS] & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [aSb] & \cdot & \cdot & \cdot & [a] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [aS] & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & [b] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [b] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & [b] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [b] & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ \hline \end{array} \right) \quad (15)$$

Результирующий граф с новыми рёбрами:



Его матрица смежности:

$$M_2 = \begin{pmatrix} [S] & [a] & [S] & \cdot & [S] & \cdot & [S] \\ \cdot & [S] & [b] & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [S] & [a] & [S] & \cdot & [S] \\ \cdot & \cdot & \cdot & [S] & [b] & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [S] & [a] & [S] \\ \cdot & \cdot & \cdot & \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] \end{pmatrix}$$

Таким образом видно, что для выбранных входных данных алгоритму достаточно двух итераций основного цикла (вторая итерация будет неполной, достаточно проверить, что результат тензорного произведения не изменился). Читателю предлагается выяснить, сколько умножений потребуется, чтобы вычислить транзитивное замыкание на первой итерации.

Теперь запустим алгоритм на второй грамматике.

[illegible]

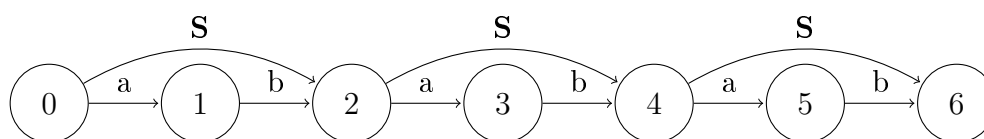
Уже сейчас можно заметить, что размер матриц, с которыми нам придётся работать, существенно увеличился, по сравнению с предыдущим вариантом. Это, конечно, закономерно, ведт в рекурсивной сети для предыдущего варианта меньше состояний, а значит и матрица смежности имеет меньший размер.

Транзитивное замыкание.

$$tc(M_3) =$$

$$= \left(\begin{array}{c|c|c|c|c} \dots\dots\dots & \begin{array}{c} [a] \\ \dots\dots\dots \\ [a] \\ \dots\dots\dots \\ [a] \\ \dots\dots\dots \end{array} & \begin{array}{c} [aS] \\ \dots\dots\dots \\ [aS] \\ \dots\dots\dots \\ [aS] \\ \dots\dots\dots \end{array} & \begin{array}{c} [aSb] \\ \dots\dots\dots \\ [aSb] \\ \dots\dots\dots \\ [aSb] \\ \dots\dots\dots \end{array} & \begin{array}{c} [aSbS] \\ \dots\dots\dots \\ [aSbS] \\ \dots\dots\dots \\ [aSbS] \\ \dots\dots\dots \end{array} \\ \hline \dots\dots\dots & \dots\dots\dots & \begin{array}{c} [S] \\ [S] \\ [S] \\ [S] \\ [S] \\ [S] \end{array} & \dots\dots\dots & \dots\dots\dots \\ \hline \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \begin{array}{c} [b] \\ [b] \\ [b] \end{array} & \dots\dots\dots \\ \hline \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \begin{array}{c} [S] \\ [S] \\ [S] \\ [S] \\ [S] \\ [S] \end{array} \\ \hline \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \end{array} \right) \quad (17)$$

Обновлённый граф



Его матрица смежности:

$$M_2 = \begin{pmatrix} [S] & [a] & \mathbf{[S]} & \cdot & \cdot & \cdot & \cdot \\ \cdot & [S] & [b] & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [S] & [a] & \mathbf{[S]} & \cdot & \cdot \\ \cdot & \cdot & \cdot & [S] & [b] & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [S] & [a] & \mathbf{[S]} \\ \cdot & \cdot & \cdot & \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] \end{pmatrix}$$

Потребуется ещё одна итерация.

[illegible]

Транзитивное замыкание.

$$M_2 = \begin{pmatrix} [S] & [a] & [S] & \cdot & \mathbf{[S]} & \cdot & \cdot \\ \cdot & [S] & [b] & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & [S] & [a] & [S] & \cdot & \mathbf{[S]} \\ \cdot & \cdot & \cdot & [S] & [b] & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & [S] & [a] & [S] \\ \cdot & \cdot & \cdot & \cdot & \cdot & [S] & [b] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & [S] \end{pmatrix}$$

И, наконец, последняя содержательная итерация.

[illegible]

Транзитивное замыкание.

3. Найти библиотеку для тензорного произведения. Реализовать алгоритм. Можно предполагать, что запросы содержат ограниченное число терминалов и нетерминалов. Провести замеры. Сравнить с матричным.

6 Сжатое представление леса разбора

Матричный алгоритм даёт нам ответ на вопрос о достижимости, но не предоставляет самих путей. Что делать, если мы хотим построить все пути, удовлетворяющие ограничениям?

Проблема в том, что множество путей может быть бесконечным. Можем ли мы предложить конечную структуру, однозначно описывающую такое множество? Вспомним, что пересечение контекстно-свободного языка с регулярным — это контекстно-свободный язык. Мы знаем, что контекстно-свободный язык можно описать контекстно-свободной грамматикой, которая конечна. Это и есть решение нашего вопроса. Осталось только научиться строить такую грамматику.

6.1 Дерево вывода

Дерево вывода цепочки в грамматике

- Корневое. Корень помечен стартовым нетерминалом.
- Упорядоченное
- Соотношение узлов и правил
- Крона — цепочка

6.2 Неоднозначные грамматики

Левосторонний и правосторонний выводы.

Существенно неоднозначные языки

6.3 Лес разбора как представление контекстно-свободной грамматики

Добавление информации в классическое дерево разбора. Координаты и промежуточные узлы.

6.4 Вопросы и задачи

1. Постройте дерево вывода цепочки $w = aababb$ в грамматике $G = \langle \{a, b\}, \{S\}, \{S \rightarrow \varepsilon \mid a S b S\}, S \rangle$.
2. Постройте все левосторонние выводы цепочки $w = ababab$ в грамматике $G = \langle \{a, b\}, \{S\}, \{S \rightarrow \varepsilon \mid a S b \mid S S\}, S \rangle$.
3. Постройте все правосторонние выводы цепочки $w = ababab$ в грамматике $G = \langle \{a, b\}, \{S\}, \{S \rightarrow \varepsilon \mid a S b \mid S S\}, S \rangle$.

4. Постройте все деревья вывода цепочки $w = ababab$ в грамматике $G = \langle \{a, b\}, \{S\}, \{S \rightarrow \varepsilon \mid a S b \mid S S\}, S \rangle$, соответствующие левосторонним выводам.
5. Постройте все деревья вывода цепочки $w = ababab$ в грамматике $G = \langle \{a, b\}, \{S\}, \{S \rightarrow \varepsilon \mid a S b \mid S S\}, S \rangle$, соответствующие правосторонним выводам.
6. Как связаны между собой леса, полученные в предыдущих двух задачах (4 и 5)? Какие выводы можно сделать из такой связи?
7. Постройте сжатое представление леса разбора, полученного в задаче 4.
8. Постройте сжатое представление леса разбора, полученного в задаче 5.
9. Предъявите контекстно-свободную грамматику существенно неоднозначного языка. Возьмите цепочку длины больше пяти, при надлежащую этому языку, и постройте все деревья вывода этой цепочки в предъявленной грамматике.
10. Постройте сжатое представление леса, полученного в задаче 9.

7 Алгоритм на основе восходящего анализа

Наша реализация [?]

7.1 Восходящий синтаксический анализ

Основы LR-анализа.

Таблицы, конфликты.

Пример автомата и таблиц.

7.2 Вопросы и задачи

1. Постройте автомат для грамматики
2. Постройте таблицу для автомата из задачи
3. В том числе дать неоднозначную грамматику
4. Запустить, построить деревья, стеки и т.д.

8 Алгоритм на основе нисходящего анализа

Комбинаторы [7] GLL [3]

Другие реализации [5]

8.1 Нисходящий синтаксический анализ

9 От CFPQ к вычислению Datalog-запросов

9.1 Datalog

Конечные Эрбрановы модели. Наименьшая неподвижная точка. $C :- d$

9.2 КС-запрос как запрос на Datalog

Покажем, что для данного графа и КС-запроса можно построить эквивалентный запрос на Datalog.

Пусть дан граф G . Граф преобразуется в набор фактов (базу данных).

Пусть есть грамматика $G: S \rightarrow a \ b \mid a \ S \ b$. Она может быть преобразована в запрос следующего вида. $s(X, Y) :- a(X, Z), b(Z, Y)$. $s(X, Y) :- a(X, Z), s(Z, W)b(W, Y)$. $? :- s(X, Y)$

Наблюдения: появились переменные, есть порядок у конъюнктов, который задаёт порядок связывания.

9.3 Обобщение GLL для вычисления Datalog-запросов

Дескриптор — состояние процесса: состояние автомата, результат проделанной работы, подстановка. Задача — найти подстановки. На каждом шаге есть набор подстановок.

Список литературы

- [1] R. Azimov and S. Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '18, pages 5:1–5:10, New York, NY, USA, 2018. ACM.
- [2] P. G. Bradford. Efficient exact paths for dyck and semi-dyck labeled path reachability (extended abstract). In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 247–253, Oct 2017.
- [3] S. Grigorev and A. Ragozina. Context-free path querying with structural representation of result. In *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia, CEE-SECR '17*, pages 10:1–10:7, New York, NY, USA, 2017. ACM.
- [4] J. Kuijpers, G. Fletcher, N. Yakovets, and T. Lindaaker. An experimental study of context-free path query evaluation methods. In *Proceedings of the 31st International Conference on Scientific and Statistical Database Management, SSDBM '19*, pages 121–132, New York, NY, USA, 2019. ACM.
- [5] C. M. Medeiros, M. A. Musicante, and U. S. Costa. Ll-based query answering over rdf databases. *Journal of Computer Languages*, 51:75 – 87, 2019.

- [6] N. Mishin, I. Sokolov, E. Spirin, V. Kutuev, E. Nemchinov, S. Gorbatyuk, and S. Grigorev. Evaluation of the context-free path querying algorithm based on matrix multiplication. In *Proceedings of the 2Nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA'19, pages 12:1–12:5, New York, NY, USA, 2019. ACM.
- [7] E. Verbitskaia, I. Kirillov, I. Nozkin, and S. Grigorev. Parser combinators for context-free path querying. In *Proceedings of the 9th ACM SIGPLAN International Symposium on Scala*, Scala 2018, pages 13–23, New York, NY, USA, 2018. ACM.