

Санкт-Петербургский государственный университет

Кафедра системного программирования

Ковалев Дмитрий Александрович

Синтаксический анализ данных,  
представленных в виде  
контекстно-свободной грамматики

Дипломная работа

Научный руководитель:  
к. ф.-м. н., \*указать должность\* Григорьев С. В.

Рецензент:  
программист \*указать место работы\* Авдюхин Д. А.

Санкт-Петербург  
2017

SAINT-PETERSBURG STATE UNIVERSITY

Chair of Software Engineering

Dmitry Kovalev

# Parsing of the data represented as context free grammar

Graduation Thesis

Scientific supervisor:  
\*position\* Semyon Grigorev

Reviewer:  
Programmer at \*company\* Avdiukhin Dmitrii

Saint-Petersburg  
2017

# Оглавление

Введение	4
1. Постановка задачи	5
2. Обзор	6
3. Алгоритм	7
Список литературы	8

# Введение

Контекстно-свободные грамматики, наряду с регулярными выражениями, повсеместно используются для решения задач, связанных с разработкой формальных языков и синтаксических анализаторов. Одним из основных достоинств контекстно-свободных грамматик является возможность задания широкого класса языков при сохранении относительной компактности представления. Благодаря данному свойству, грамматики также представляют интерес в такой области информатики, как кодирование и сжатие данных. В частности, существует ряд алгоритмов, позволяющих производить сжатие текстовой информации, используя в качестве конечного [8] или промежуточного [3] представления контекстно-свободную грамматику (grammar-based compression).

Использование компактного представления текста может быть актуально для некоторых задач биоинформатики. Примером такой задачи является анализ метагеномных сборок — помеченных графов большого размера: порядка  $10^6$  ребер и  $10^6$  вершин. Метками на ребрах графа являются строки над алфавитом нуклеотидных символов. В магистерской диссертации Анастасии Рагозиной [9] был предложен алгоритм, позволяющий искать в метагеномных сборках шаблоны РНК, задаваемые при помощи контекстно-свободной грамматики. Производительность данного алгоритма, предположительно, может быть увеличена за счет контекстно-свободного сжатия меток на ребрах графа и трансформации его в грамматику, представленную в расширенной форме Бэкуса-Наура.

Задача поиска шаблонов при использовании такого представления метагеномной сборки формулируется следующим образом: необходимо найти все строки, принадлежащие перечению языков, задаваемых грамматикой шаблона и грамматикой сборки. В общем случае такая задача неразрешима, так как сводится к проверке пересечения двух произвольных контекстно-свободных языков на пустоту. Поэтому необходимо выставить ограничения и вообще...

# 1. Постановка задачи

Целью данной работы является разработка алгоритма синтаксического анализа данных, представленных в виде контекстно-свободной грамматики.

Данная цель приводит к необходимости решения задач, связанных с проверкой пустоты пересечения двух контекстно-свободных языков. Известно, что в общем случае данная проблема неразрешима. В связи с этим, для достижения цели задачи были сформулированы следующим образом.

- Изучить существующие подходы к анализу данных, представленных в виде КС-грамматик.
- Определить ограничения, при которых синтаксический анализ такого представления является разрешимой задачей.
- Разработать алгоритм синтаксического анализа контекстно-свободного представления данных с учетом поставленных ограничений.
- Реализовать алгоритм в рамках проекта YaccConstructor.
- Доказать завершаемость алгоритма.
- Провести тестирование и апробацию.

## 2. Обзор

Пусть  $G$  — произвольная КС-грамматика,  $M$  — конечный автомат. Тогда задача проверки

- включения языков ( $L(M) \subseteq L(G)$ ) — неразрешима
- пустоты пересечения ( $L(M) \cap L(G) = \emptyset$ ) — разрешима (т.к. в пересечении не более чем КС-язык) за полиномиальное время [6]
- регулярности языка  $L(G)$  — неразрешима [5]

Если использовать представление регулярного языка  $L(M)$  в виде КС-грамматики  $G_r$ , то задача проверки пустоты пересечения ( $L(G_r) \cap L(G) = \emptyset$ ) становится немного интереснее: если  $G_r$

- нерекурсивная — задача из PSPACE [7] (точнее результата нет (я не нашел, по крайней мере))
- лево- или праволинейная — ничего не известно (см. последний абзац заключения из [7])
- принадлежит еще более широкому классу — тем более ничего не известно

Еще немного про вложенную рекурсию и регулярность языка. Грамматика без вложенной рекурсии (NSE) порождает регулярный язык [4] (обратное тоже верно, для регулярного языка можно построить NSE грамматику, т.к. праволинейная, например, — частный случай NSE). Существует алгоритм, который позволяет проверять грамматику на наличие вложенной рекурсии за полином [2]. Однако, грамматика с вложенной рекурсией тоже может порождать регулярный язык [1], поэтому задача о проверке регулярности языка, порождаемого КС-грамматикой, остается неразрешимой.

### 3. Алгоритм

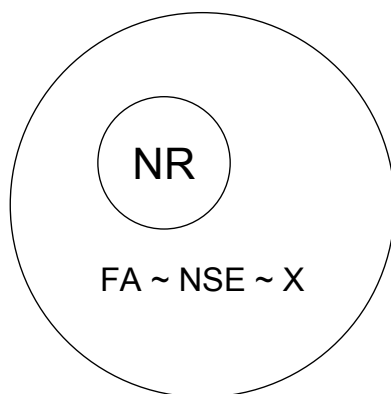
Мы пытаемся решать следующую задачу: пусть  $G_1 = (N, T, S, P)$  — произвольная КС-грамматика,  $G_2$  — NSE КС-грамматика. Алгоритм принимает на вход два рекурсивных автомата,  $M_1$  и  $M_2$ , построенных по грамматикам  $G_1$  и  $G_2$  соответственно, при этом в автомате  $M_2$  левые/правые рекурсивные вызовы заменены на циклы, как в обычном конечном автомате.

Результатом работы алгоритма являются тройки вида  $(X, n_1, n_2)$ , где  $X \in N$ ,  $n_1, n_2$  — номера состояний автомата  $M_2$ . Для каждой из таких троек выполняется следующее утверждение:  $\exists \omega \in T^*$  такая, что  $X \rightarrow^* \omega$  в  $G_1$  и  $\omega \in L(M')$ , где  $M'$  — рекурсивный автомат, полученный из  $M_2$  путем замены начального и конечного состояния на  $n_1$  и  $n_2$  соответственно.

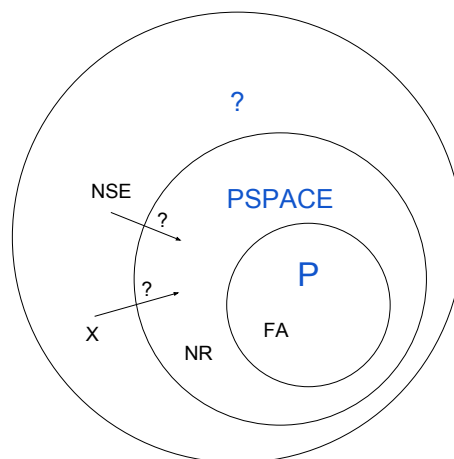
Получая такие результаты, мы, по сути, отвечаем на вопрос о проверке пустоты пересечения КС-языка и регулярного, представленных в необычных абстракциях. Для КС-языка мы используем рекурсивный автомат, а для регулярного — нечто среднее между конечным автоматом и NSE грамматикой (это нечто все еще использует стек, но только для обработки нерекурсивных вызовов). Такое представление эквивалентно по выразительности NSE и, следовательно, FA (см. рис. 1a). Но неизвестно, к какому классу сложности относится задача (и разрешима ли вообще) о проверке пустоты пересечения регулярного языка, представленного в данной форме, с КС-языком (см. рис. 1b).

**Theorem 1** *Завершаемость. Алгоритм завершает работу за конечное число шагов для произвольных входных данных*

**Theorem 2** *Корректность. ???*



(a) Выразительность языка, задаваемого абстракцией



(b) Класс сложности для задачи о проверке пустоты пересечения с КС-языком

Рис. 1: Красивые круги. Здесь NR — нерекурсивная грамматика, FA — конечный автомат, NSE — грамматика без вложенной рекурсии, X — наше представление

## Список литературы

- [1] Andrei Stefan, Chin Wei-Ngan, Cavadini Salvador Valerio. Self-embedded context-free grammars with regular counterparts // Acta Informatica. — 2004. — Vol. 40, no. 5. — P. 349–365. — Access mode: <http://dx.doi.org/10.1007/s00236-003-0133-8>.
- [2] Anselmo Marcella, Giammarresi Dora, Varricchio Stefano. Finite Automata and Non-self-embedding Grammars // Proceedings of the 7th International Conference on Implementation and Application of Automata. — CIAA'02. — Berlin, Heidelberg : Springer-Verlag, 2003. — P. 47–56. — Access mode: <http://dl.acm.org/citation.cfm?id=1756384.1756390>.
- [3] Arimura Mitsuharu. A grammar-based compression using a variation of Chomsky normal form of context free grammar // 2016 International Symposium on Information Theory and Its Applications (ISITA). — 2016.



- [4] Chomsky Noam. A note on phrase-structure grammars // Information and Control. — 1959. — Vol. 2. — P. 393 – 395.
- [5] Greibach Sheila. A note on undecidable properties of formal languages // Mathematical systems theory. — 1968. — Vol. 2, no. 1. — P. 1–6. — Access mode: <http://dx.doi.org/10.1007/BF01691341>.
- [6] Hunt III Harry B., Rosenkrantz Daniel J., Szymanski Thomas G. On the Equivalence, Containment, and Covering Problems for the Regular and Context-free Languages // J. Comput. Syst. Sci. — 1976. — Apr. — Vol. 12, no. 2. — P. 222–268. — Access mode: [http://dx.doi.org/10.1016/S0022-0000\(76\)80038-4](http://dx.doi.org/10.1016/S0022-0000(76)80038-4).
- [7] Nederhof Mark-Jan, Satta Giorgio. The Language Intersection Problem for Non-recursive Context-free Grammars // Inf. Comput. — 2004. — Aug. — Vol. 192, no. 2. — P. 172–184. — Access mode: <http://dx.doi.org/10.1016/j.ic.2004.03.004>.
- [8] Nevill-Manning Craig G., Witten Ian H. Identifying Hierarchical Structure in Sequences: A Linear-time Algorithm // J. Artif. Int. Res. — 1997. — Sep. — Vol. 7, no. 1. — P. 67–82. — Access mode: <http://dl.acm.org/citation.cfm?id=1622776.1622780>.
- [9] Ragozina Anastasiya. GLL-based relaxed parsing of dynamically generated code. — 2016.