

О некоторых частных случаях задачи поиска путей с контекстно-свободными ограничениями

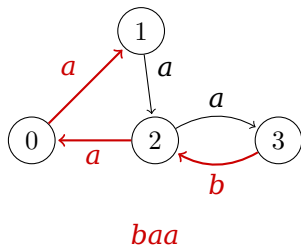
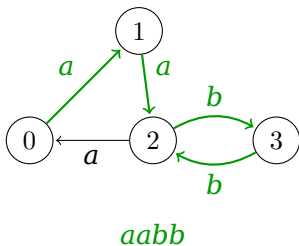
Олемская Александра Витальевна
научный руководитель: к.ф.-м.н. С.В. Григорьев

НИУ ВШЭ — Санкт-Петербург

9 июня 2021 г.

Задача поиска путей с контекстно-свободными ограничениями

- Ориентированный граф с метками G
- Контекстно-свободная грамматика \mathcal{G}
- Найти все пары вершин u, v , такие что есть путь $u \rightsquigarrow v$, на котором читается слово, выводимое грамматикой
- Пример
Язык слов вида $a^n b^n$ (задаётся грамматикой $S \rightarrow aSb \mid \varepsilon$)



- Анализ данных, использующих графовую модель представления
 - Запросы к графовым базам данных¹
 - Работа с биомедицинскими данными²
 - Анализ социальных сетей³
- Статический анализ⁴
 - межпроцедурный анализ потока данных (interprocedural dataflow analysis)
 - анализ указателей (points-to analysis)⁵
 - анализ формы (shape analysis)⁶

¹Medeiros и др., «Efficient Evaluation of Context-Free Path Queries for Graph Databases», 2018.

²Sevon и Eronen, «Subgraph Queries by Context-free Grammars», 2008.

³Chaudhary и Faisal, «Role of graph databases in social networks», 2016.

⁴Reps, «Program analysis via graph reachability», 1998.

⁵Xu и др., «Scaling CFL-reachability-based points-to analysis using context-sensitive must-not-alias analysis», 2009.

⁶Rinetzky и др., «Modular shape analysis for dynamically encapsulated programs», 2007.

Существующие решения

- Много разных, по большей части, основанных на разных техниках синтаксического анализа⁷⁸
- Асимптотика всех⁹ решений $\mathcal{O}(n^3)$
- Не достаточно эффективны для использования на практике¹⁰
- Есть условная нижняя оценка¹¹: вряд ли существует комбинаторное решение за $\mathcal{O}(n^{3-\varepsilon})$
- Осмысленно рассматривать частные случаи

⁷Melski и Reps, «Interconvertibility of a class of set constraints and context-free-language reachability», 2000.

⁸Hellings, «Path Results for Context-free Grammar Queries on Graphs», 2015.

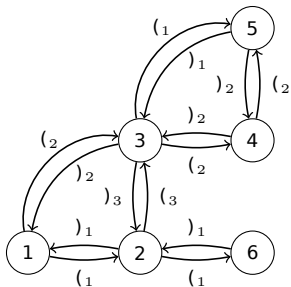
⁹Некоторые можно ускорить в $\log n$ раз методом четырёх русских

¹⁰Kuijpers и др., «An Experimental Study of Context-Free Path Query Evaluation Methods», 2019.

¹¹Heintze и McAllester, «On the cubic bottleneck in subtyping and flow analysis», 1997.

Частные случаи

- Язык Дика \mathcal{D}_k : язык ПСП на k типах скобок
- Используется в стат. анализе
- $\mathcal{O}(n^\omega \log^2 n)$ для языка Дика на одном типе скобок¹²
- $\mathcal{O}(m + n\alpha(n))$ для языка Дика и двунаправленных графов¹³
- И другие¹⁴
- Проблема: слишком разные подходы, нельзя переиспользовать



¹²Mathiasen и Pavlogiannis, «The Fine-Grained and Parallel Complexity of Andersen's Pointer Analysis», 2021.

¹³Chatterjee и др., «Optimal Dyck Reachability for Data-Dependence and Alias Analysis», 2017.

¹⁴Chaudhuri, «Subcubic Algorithms for Recursive State Machines», 2008.

Цель:

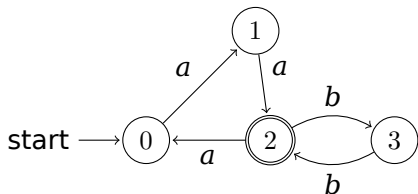
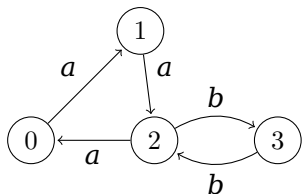
Получить решения для частных случаев в задаче поиска путей с контекстно-свободными ограничениями, основываясь на едином подходе

Задачи:

- Выбрать единый подход
- Построить на его основе решения для следующих частных случаев:
 - Язык Дика и двунаправленные графы
 - Язык Дика на одном типе скобок
- Применить выбранный подход для решения задачи достижимости для смешанного языка Дика и двунаправленных графов

Алгоритм, основанный на пересечении языков

- Описан в¹⁵ и¹⁶
- Входной граф — автомат, задающий язык $L(G, u, v)$ слов, читаемых на пути между парой вершин



- Из u в v есть путь, на котором читается слово из $L(\mathcal{G}) \Leftrightarrow L(G, u, v) \cap L(\mathcal{G})$ непусто

¹⁵Chaudhuri, «Subcubic Algorithms for Recursive State Machines», 2008.

¹⁶Orachev и др., «Context-Free Path Querying by Kronecker Product», 2020.

Алгоритм, основанный на пересечении языков II

- Пересечение двух языков — прямое произведение автоматов
- Граф — НКА, КС-язык — РКА (рекурсивный автомат) \Rightarrow прямое произведение — тоже РКА
- Непустота языка \Leftrightarrow из стартового состояния достижимо конечное \Rightarrow решаем задачу достижимости для РКА
- Инкрементальное транзитивное замыкание (добавление рёбер + проверка достижимости)
- В общем случае решается за $\mathcal{O}(n^3)$

Будем видоизменять данный алгоритм, чтобы получить более быстрые решения

- Вместо обычного транзитивного замыкания поддерживаем неориентированное
- На неориентированном графе отношение достижимости = отношение “лежать в одной компоненте связности”
- Поддерживаем компоненты и проводим новые рёбра с помощью СНМ
- Работает за $\mathcal{O}(n^2\alpha(n))$
- (Теорема) Работает корректно на двунаправленных графах и языке Дика






Язык Дика на одном типе скобок





- Вместо добавления рёбер по одному, добавляем сразу все и пересчитываем транзитивное замыкание
- Работает за $\mathcal{O}(n^\omega \cdot \# \text{итераций})$ ¹⁷
- Пользуемся фактом¹⁸, что если есть путь, то есть и путь длины $\mathcal{O}(n^2)$
- Строим специальный РКА, для которого число итераций будет $\mathcal{O}(\log^2 n)$
- Размер РКА будет не константа, а $\mathcal{O}(\log n)$, получаем $\mathcal{O}(n^\omega \log n)$ на итерацию
- Итого $\mathcal{O}(n^\omega \log^3 n)$






¹⁷ $\mathcal{O}(n^\omega)$ — время перемножения матриц, где $\omega < 2.373$

¹⁸Deleage и Pierre, «The Rational Index of the Dyck Language D_1^* », 1986.

- Разработан общий подход к созданию частных решений для задачи поиска путей с контекстно-свободными ограничениями
- Получен алгоритм, работающий за $\mathcal{O}(n^{2\alpha(n)})$ для языка Дика и двунаправленных графов
- Получен алгоритм, работающий за $\tilde{\mathcal{O}}(n^\omega)$ для языка Дика на одном типе скобок
- Получен алгоритм, работающий за $\mathcal{O}(n^4)$ для смешанного языка Дика $\mathcal{D}_1 \odot \mathcal{D}_1$ и двунаправленных графов

-  Chatterjee, Krishnendu, Bhavya Choudhary и Andreas Pavlogiannis. «Optimal Dyck Reachability for Data-Dependence and Alias Analysis». В: *Proc. ACM Program. Lang.* 2.POPL (дек. 2017).
-  Chaudhary, Anoop и Abdul Faisal. «Role of graph databases in social networks». В: (июнь 2016).
-  Chaudhuri, Swarat. «Subcubic Algorithms for Recursive State Machines». В: POPL '08. San Francisco, California, USA: Association for Computing Machinery, 2008, с. 159—169.
-  Deleage, Jean-Luc и Laurent Pierre. «The Rational Index of the Dyck Language D_1^* ». В: *Theor. Comput. Sci.* 47.3 (нояб. 1986), с. 335—343.
-  Heintze, N. и D. McAllester. «On the cubic bottleneck in subtyping and flow analysis». В: *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science.* 1997, с. 342—351.

-  Hellings, Jelle. «Path Results for Context-free Grammar Queries on Graphs». В: *CoRR abs/1502.02242* (2015). arXiv: 1502.02242.
-  Kuijpers, Jochem и др. «An Experimental Study of Context-Free Path Query Evaluation Methods». В: *SSDBM '19*. Santa Cruz, CA, USA: Association for Computing Machinery, 2019, с. 121—132.
-  Mathiasen, Anders Alnor и Andreas Pavlogiannis. «The Fine-Grained and Parallel Complexity of Andersen's Pointer Analysis». В: *Proc. ACM Program. Lang.* 5.POPL (январь 2021).
-  Medeiros, Ciro M., Martin A. Musicante и Umberto S. Costa. «Efficient Evaluation of Context-Free Path Queries for Graph Databases». В: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. SAC '18. Pau, France: Association for Computing Machinery, 2018, с. 1230—1237.

-  Melski, David и Thomas Reps. «Interconvertibility of a class of set constraints and context-free-language reachability». В: *Theoretical Computer Science* 248.1 (2000). PEPM'97, с. 29—98.
-  Orachev, Egor и др. «Context-Free Path Querying by Kronecker Product». В: *Advances in Databases and Information Systems*. Под ред. Jérôme Darmont, Boris Novikov и Robert Wrembel. Cham: Springer International Publishing, 2020, с. 49—59.
-  Reps, Thomas. «Program analysis via graph reachability». В: *Information and Software Technology* 40.11 (1998), с. 701—726.
-  Rinetzky, Noam и др. «Modular shape analysis for dynamically encapsulated programs». В: *European Symposium on Programming*. Springer. 2007, с. 220—236.
-  Sevon, Petteri и Lauri Eronen. «Subgraph Queries by Context-free Grammars». В: *Journal of Integrative Bioinformatics* 5.2 (2008), с. 157—172.



Xu, Guoqing, Atanas Rountev и Manu Sridharan. «Scaling CFL-reachability-based points-to analysis using context-sensitive must-not-alias analysis». В: *European Conference on Object-Oriented Programming*. Springer. 2009, с. 98—122.

Общая схема решений, основанных на пересечении языков

- Построить РКА для пересечения входного графа и входного КС-языка
- Решить задачу достижимости для полученного РКА

```
while ( $\exists$  новый путь  $u \rightsquigarrow v$ ) do  
  | if ( $u$  — стартовая  $\wedge v$  — конечная) then  
  |   | Добавить новые рекурсивные рёбра  
  | end  
  | Обновить отношение достижимости  
end
```

Добавление рёбер + проверка достижимости \Rightarrow Задача *инкрементального транзитивного замыкания*

- Ответ на запрос (u, v) — да \Leftrightarrow в РКА произведении есть путь из стартового состояния u в конечное состояние v

Частные случаи (было/стало)

Частный случай	Было	Стало
Язык Дика \mathcal{D}_k и двунаправленные графы	$\mathcal{O}(m + n\alpha(n))$	$\mathcal{O}(m^*\alpha(n))$
Неориентированные графы и специальные виды грамматик	—	$\mathcal{O}(m^*\alpha(n))$
Язык Дика \mathcal{D}_1	$\mathcal{O}(n^\omega \log^2 n)$	$\mathcal{O}(n^\omega \log^3 n)$
Смешанный язык Дика $\mathcal{D}_1 \odot \mathcal{D}_1$	$\mathcal{O}(n^7)$	$\mathcal{O}(n^4)$

где n — число вершин в графе,
 m — число рёбер, m^* — число пар вершин в ответе.

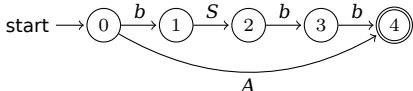
Рекурсивные автоматы (РКА)

- Неформально, это набор компонент, каждая из которых представляет собой НКА, на рёбрах которого могут быть “рекурсивные вызовы” других компонент.
- Напишем грамматику для языка слов вида $b^n a^m b^{2n}$:

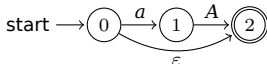
$$L: S \rightarrow bSbb \mid A$$

$$A \rightarrow aA \mid \varepsilon$$

- РКА по грамматике обычно строится следующим образом: создаётся компонента для каждого нетерминала, в которой проводятся пути, соответствующие продукциям.



(a) Компонента S



(b) Компонента A