Запросы к графовым базам данных в терминах формальных языков

В основном про RPQ и немного больше

Николай Пономарев

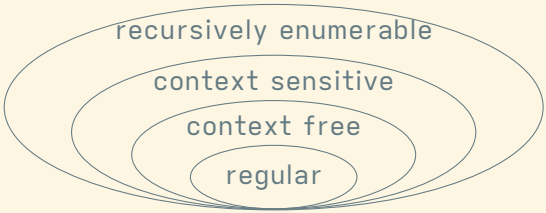
Математико-механический факультет СПбГУ

8 октября 2024 г.

Иерархия Хомского

Небольшое напоминание:

- Рекурсивно перечислимые языки. Продукции вида: $\gamma \to \alpha$
- Контекстно-зависимые языки. Продукции вида: $\alpha A \beta \to \alpha \gamma \beta$
- Контекстно-свободные языки. Продукции вида: $A \to \alpha \gamma \beta$
- Регулярные языки. Продукции вида: $A \to a$, $A \to aB$



Обозначения: a — терминал, A, B — нетерминалы, α, β, γ — цепочки терминалов и/или нетерминалов

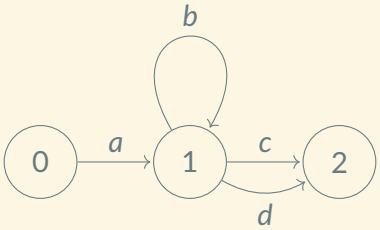
Наш главный объект изучения — граф

Определение (из [She+21])

Реберно-меченный ориентированный мультиграф — это $G = (V, E, \Sigma)$, где

- V — конечное множество вершин, обычно от 0 до $|V| - 1$
- $E \subseteq V \times \Sigma \times V$ — конечное множество ребер
- Σ — конечный алфавит меток

Знакомьтесь, реберно-меченный мультиорграф



$$\mathcal{G} = (V = \{0, 1, 2\}, E = \{(0, a, 1), (1, b, 1), (1, c, 2), (1, d, 2)\}, \Sigma = \{a, b, c, d\})$$

Ещё немного определений про графы

Определение

Путём π в графе $\mathcal{G} = (V, E, \Sigma)$ называется последовательность $e_0, e_1, \dots, e_{n-1}$, где $e_i = (v_i, \sigma_i, u_i) \in E$, и для всех e_i, e_{i+1} выполнено $u_i = v_{i+1}$. Будем обозначать путь из v в u как $v \pi u$.

Определение

Словом, образованным путём $\pi = (v_0, \sigma_0, u_0), (v_1, \sigma_1, u_1), \dots, (v_{n-1}, \sigma_{n-1}, u_{n-1})$, будем называть конкатенацию всех меток этого пути:

$$\lambda(\pi) = \sigma_0 \sigma_1 \dots \sigma_{n-1}$$

А теперь про формальные языки

Определение (из [MW95])

Пусть Σ — конечный алфавит, не содержащий $\{\varepsilon, \varnothing, (,)\}$. Регулярное выражение R над Σ определено следующим образом:

- Пустая строка ε, пустое множество $\varnothing$, и все $a \in \Sigma$ являются регулярными выражениями
- Если A и B — регулярные выражения, то $(A + B)$ (альтернатива), AB (конкатенация), A^* (звезда Клини) тоже регулярные выражения
- Ничего другого регулярным выражением не является

И ещё про формальные языки

Определение

Язык $\mathcal{L}(R)$, описываемый R, определяется следующим образом:

1. $\mathcal{L}(\varepsilon) = \{\varepsilon\}$
2. $\mathcal{L}(\varnothing) = \varnothing$
3. $\mathcal{L}(a) = \{a\}\ \forall a \in \Sigma$
4. $\mathcal{L}(A + B) = \mathcal{L}(A) \cup \mathcal{L}(B) = \{w : w \in \mathcal{L}(A) \vee w \in \mathcal{L}(B)\}$
5. $\mathcal{L}(AB) = \mathcal{L}(A)\mathcal{L}(B) = \{w_1 w_2 : w_1 \in \mathcal{L}(A) \wedge w_2 \in \mathcal{L}(B)\}$
6. $\mathcal{L}(A^*) = \bigcup_{k=0}^{\infty} \mathcal{L}(A)^k$, где $\mathcal{L}(A)^0 = \{\varepsilon\}$ и $\mathcal{L}(A)^k = \mathcal{L}(A)^{k-1}\mathcal{L}(A)$

Постановка задачи RPQ

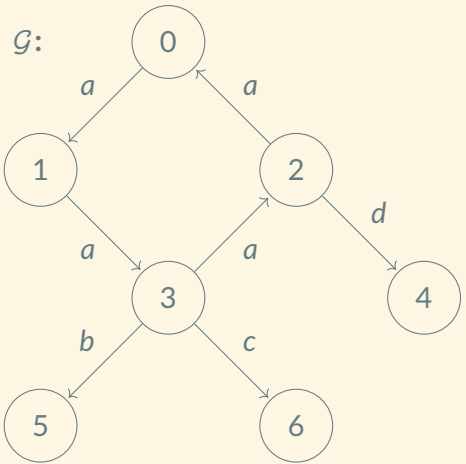
Определение

Пусть $\mathcal{G} = (V, E, \Sigma)$ — реберно-меченный мультиорграф, R — регулярное выражение над Σ. Тогда решением *задачи достижимости с ограничениями в терминах регулярных языков* для $\mathcal{G}$ и R является множество

$$\left\{(u, v) \in V \times V : \exists \text{ путь } \pi \text{ в } \mathcal{G} \text{ из } u \text{ в } v : \lambda(\pi) \in \mathcal{L}(R)\right\}$$

- На английском, Regular Path Query (RPQ)
- Часто назначают стартовые и конечные вершины

Пример RPQ (из [NS16])



$$R_1 = (a + aa)(b + d)$$
$$\mathcal{L}(R_1) = \{ab, aab, ad, aad\}$$
$$\mathrm{RPQ}(\mathcal{G}, R_1) = \{(0,5), (1,5), (1,4), (3,4)\}$$

$$R_2 = (aaaa)^*(b + c + d)$$
$$\mathcal{L}(R_2) = \left\{ w_1 w_2 : w_1 \in \bigcup_{k=0}^{\infty} (aaaa)^k \wedge w_2 \in \{b, c, d\} \right\}$$
$$\mathrm{RPQ}(\mathcal{G}, R_2) = \{(3,5), (3,6), (2,4)\}$$

Как это вычислять?

1. Пересечение конечных автоматов
   - Обычно решает задачу для всех вершин
   - Реализация почти очевидна из названия, немного есть в [She+21]
2. Синхронный обход в ширину конечного автомата и графа
   - Часто решает задачу для конкретных стартовых, конечных вершин
   - Способ умнее, почитать можно в [Ele+20]

Кто это поддерживает? I

Графовые базы данных

- Начало развития теории — конец 80-х (см. [Bon+18, с. 35])
- Ответственный за популярность — Neo4j (2007)
- Сейчас главное — язык запросов
  - openCypher [Neo]: Neo4j, Amazon Neptune, ArcadeDB, Redis Graph, VK Tarantool
  - SPARQL [W3C]: Amazon Neptune, Eclipse RDF4J, Apache Jena
    - Стандарт W3C
  - Apache TinkerPop's Gremlin [Apa]: Neo4j, JanusGraph, Amazon Neptune, ArcadeDB
  - GQL [ISO]: на данный момент ∅
    - Стандарт ISO 2024 года!
    - Унификация всего, что выше и не только

Кто это поддерживает? II

Реляционные базы данных

- Стандарт SQL/PGQ (property graph queries)
- Разработан Oracle, принят в ISO SQL:2023 [MB23]
- Посмотреть как это выглядит можно в [Deu+21]

Области применения RPQ (по [GA24; Bon+18])

- Графы топологии сетей
- Социальные сети
- Биология

А можно круче?

Если хочется ещё более мощной системы, то есть

1. CFPQ
2. Datalog

CFPQ

- Используем вместо регулярных языков КС-языки
- Внезапно, оно довольно быстро работает ([Мур24])
- Но нет спроса у пользователей
- Известное применение — статический анализ ([Yam+14])

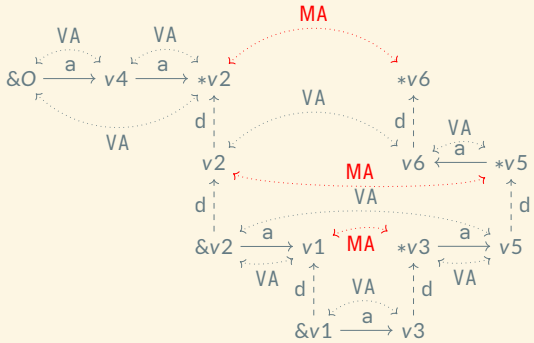


Рисунок из [Кут23]

Datalog

- Формально, это уже про дедуктивные БД
- Datalog — де факто стандарт
- В нём выразим RPQ ([Gre+13])
- И даже больше

```
r1 reachable(X,Y) :- link(X,Y).
r2 reachable(X,Y) :- link(X,Z), reachable(Z,Y).
query(X,Y) :- reachable(X,Y).
```

Вопросы к зачёту

1. Постановка задачи RPQ. Пример решения задачи для некоторого графа и регулярного выражения.
2. Примеры языков запросов и ПО, использующего их. Области использования RPQ.
3. CFPQ и Datalog, их отношение к RPQ.

Источники I

[Apa] Apache TinkerPop. *Apache TinkerPop: Gremlin*. URL: https://tinkerpop.apache.org/gremlin.html (дата обр. 06.10.2024).

[Bon+18] Angela Bonifati и др. *Querying Graphs*. Synthesis Lectures on Data Management. Cham: Springer International Publishing, 2018. ISBN: 978-3-031-00736-1 978-3-031-01864-0. DOI: 10.1007/978-3-031-01864-0. URL: https://link.springer.com/10.1007/978-3-031-01864-0 (дата обр. 30.09.2024).

[Deu+21] Alin Deutsch и др. *Graph Pattern Matching in GQL and SQL/PGQ*. 12 дек. 2021. DOI: 10.48550/arXiv.2112.06217. arXiv: 2112.06217[cs]. URL: http://arxiv.org/abs/2112.06217 (дата обр. 06.10.2024).

[Ele+20] Marton Elekes и др. «A GraphBLAS solution to the SIGMOD 2014 Programming Contest using multi-source BFS». В: *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. 2020 IEEE High Performance Extreme Computing Conference (HPEC). Waltham, MA, USA: IEEE, 22 сент. 2020, с. 1—7. ISBN: 978-1-72819-219-2. DOI: 10.1109/HPEC43674.2020.9286186. URL: https://ieeexplore.ieee.org/document/9286186/ (дата обр. 31.05.2024).

[GA24] Roberto García и Renzo Angles. «Path Querying in Graph Databases: A Systematic Mapping Study». В: *IEEE Access* 12 (2024). Conference Name: IEEE Access, с. 33154—33172. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3371976. URL: https://ieeexplore.ieee.org/document/10456906 (дата обр. 30.09.2024).

Источники II

[Gre+13] Todd J. Green и др. «Datalog and Recursive Query Processing». В: *Foundations and Trends® in Databases* 5.2 (20 нояб. 2013). Publisher: Now Publishers, Inc., с. 105—195. ISSN: 1931-7883, 1931-7891. DOI: 10.1561/1900000017. URL: https://www.nowpublishers.com/article/Details/DBS-017 (дата обр. 07.10.2024).

[ISO] ISO/IEC 39075. *Information technology — Database languages — GQL*. URL: https://www.iso.org/standard/76120.html (дата обр. 06.10.2024).

[MB23] Jim Melton и Jörn Bartels. *Information technology — Database language — SQL*. Июнь 2023. URL: https://www.iso.org/standard/76583.html (дата обр. 06.10.2024).

[MW95] Alberto O. Mendelzon и Peter T. Wood. «Finding Regular Simple Paths in Graph Databases». В: *SIAM Journal on Computing* 24.6 (дек. 1995). Publisher: Society for Industrial and Applied Mathematics, с. 1235—1258. ISSN: 0097-5397. DOI: 10.1137/S009753979122370X. URL: https://epubs.siam.org/doi/10.1137/S009753979122370X (дата обр. 05.10.2024).

[Neo] Neo4j. *openCypher · openCypher*. URL: https://opencypher.org/ (дата обр. 06.10.2024).

[NS16] Maurizio Nolé и Carlo Sartiani. «Regular Path Queries on Massive Graphs». В: *Proceedings of the 28th International Conference on Scientific and Statistical Database Management*. SSDBM '16. New York, NY, USA: Association for Computing Machinery, 18 июля 2016, с. 1—12. ISBN: 978-1-4503-4215-5. DOI: 10.1145/2949689.2949711. URL: https://doi.org/10.1145/2949689.2949711 (дата обр. 27.05.2024).

Источники III

[She+21] Ekaterina Shemetova и др. *One Algorithm to Evaluate Them All: Unified Linear Algebra Based Approach to Evaluate Both Regular and Context-Free Path Queries*. 26 марта 2021. DOI: 10.48550/arXiv.2103.14688. arXiv: 2103.14688[cs]. URL: http://arxiv.org/abs/2103.14688 (дата обр. 31.05.2024).

[W3C] W3C. *SPARQL 1.1 Query Language*. URL: https://www.w3.org/TR/sparql11-query/ (дата обр. 06.10.2024).

[Yam+14] Fabian Yamaguchi и др. «Modeling and Discovering Vulnerabilities with Code Property Graphs». В: *2014 IEEE Symposium on Security and Privacy*. 2014 IEEE Symposium on Security and Privacy. ISSN: 2375-1207. Май 2014, с. 590—604. DOI: 10.1109/SP.2014.44. URL: https://ieeexplore.ieee.org/document/6956589 (дата обр. 07.10.2024).

[Кут23] Владимир Кутуев. «Experimental investigation of context-free-language reachability algorithms as applied to static code analysis». В: (2023). Accepted: 2023-07-26T12:44:25Z. URL: https://dspace.spbu.ru/handle/11701/42628 (дата обр. 07.10.2024).

[Мур24] Илья Муравьев. «Optimisation of the context-free language reachability matrix-based algorithm». В: (2024). Accepted: 2024-07-25T11:49:47Z. URL: https://dspace.spbu.ru/handle/11701/46282 (дата обр. 07.10.2024).