Слабоструктурированные данные и поиск на основе онтологий

Афонин С.А.[1], Горелов С.С.[1], Хазова Е.Е.[1]

[1]*НИИ механики МГУ, Мичуринский пр., д. 1, г. Москва, 119192, Россия.*

serg@msu.ru, volerog@gmail.com, celena@s2s.msu.ru

Аннотация. *В работе представлен обзор основных алгоритмов и методов управления слабоструктурированными данными, представленных в виде помеченных ориентированных графов, и рассматриваются возможные их применения в области информационного поиска с использованием метаданных и онтологий.*

Ключевые слова: регулярный путевой запрос, материализованное представление, онтология, дискриптивная логика.

1 Введение

Разработка методов интеграции разнородных информационных источников является одной из наиболее актуальных задач в современной теории баз данных и информационных систем [9]. Для ее решения применяются различные подходы, начиная с интеграции на основе контекстного поиска и заканчивая построением виртуальных реляционных баз данных. Промежуточное положение занимают методы, использующие понятие слабоструктурированных данных [4]. Модель данных, использующаяся в этом подходе, представляет данные в виде ориентированного графа с помеченными ребрами и разрабатывалась для унификации формата передачи данных между разнородными приложениями.Графическое представление данных обладает большой выразительной силой, поскольку практически любая структура данных может быть представлена в таком виде при условии правильного выбора множества меток, однако поиск данных в этом случае имеет высокую алгоритмическую сложность. Большинство из языков запросов, предложенных для этой модели данных, приводит к NP-полным задачам. Сложность вычисления запросов для графовых моделей данных и широкое распространение языка разметки XML во многом способствовали смещению исследований в сторону древовидных моделей данных. Однако многочисленные результаты, полученные за время многолетних исследований графовых моделей данных, могут оказаться востребованными в области информационного поиска с использованием онтологий и метаданных. Во-первых, онтологии отражают взаимосвязи между объектами и понятиями реального мира, что естественным образом приводит к возникновению ориентированных графов с помеченными ребрами. А во-вторых, графовая структура возникает при поиске в гипертекстовых документах. После идентификации в документах понятий, представленных в онтологии, между ними возникают дополнительные связи, порожденные гипертекстовой структурой.

Работа имеет следующую структуру. В разделе 2 вводятся основные понятия модели слабоструктурированных данных. Методы вычисления и оптимизации регулярных путевых запросов рассматриваются в разделе 3. В разделе 4 обсуждается связь между рассматриваемыми задачами вычисления запросов и задачами, возникающими при вычислении запросов к онтологиям, заданных с использованием дискриптивной логики. В заключении кратко формулируются возможные направления дальнейшей работы.

2 Модель слабоструктурированных данных

Множество всех регулярных языков над алфавитом Σ будем обозначать через $\mathrm{Reg}(\Sigma)$. *Полуструктурированным документом* называется ориентированный граф с помеченными ребрами $D = \langle \mathbf{V}, \Sigma, \mathbf{E}, V_0 \rangle$, где $\mathbf{V}$ — множество вершин графа, Σ — конечное множество меток ребер, $\mathbf{E} \subseteq \mathbf{V} \times \Sigma \times \mathbf{V}$ — множество Σ-помеченных ребер, $V_0 \subseteq \mathbf{V}$ – корневые вершины.

Корневые вершины отражают деление документа на составные части, что позволяет не делать различия между отдельным документом и коллекцией документов.

Языки запросов к базам полуструктурированных данных проектируются с учетом главной особенности таких данных — отсутствия единой, строго определенной, схемы. Семантически одинаковые фрагменты базы данных могут иметь различную структуру данных. Язык запросов должен содержать специальные механизмы, обеспечивающие получение «осмысленного» результата даже при несоответствии структуры данных в различных фрагментах базы данных.

Наибольшее распространение получили языки запросов к полуструктурированным данным, основанные на поиске по шаблону [10, 8, 7, 1], которые в качестве базового механизма используют *регулярные путевые выражения*. Регулярные путевые выражения являются, по сути, регулярными языками. В качестве языка запросов к полуструктурированным данным будем рассматривать конъюнктивные регулярные путевые запросы.

Определение 1. Конъюнктивным регулярным путевым запросом *(CRPQ) называется ориентированный помеченный граф* $Q = \langle X, \mathrm{Reg}(\Sigma), E_Q, X_0 \rangle$, *где* X *— множество вершин запроса (переменных),* $\mathrm{Reg}(\Sigma)$ *обозначает множество всех регулярных языков над алфавитом* Σ, $E_Q \subset X \times \mathrm{Reg}(\Sigma) \times X$ *— множество помеченных ребер,* $X_0 \subseteq X$ *— корневые вершины запроса.*

Документ соответствует запросу, если существует отображение между вершинами запроса и вершинами документа при котором образы смежных вершин запроса соединены по крайней мере одним путем, метки которого образуют слово из регулярного языка, приписанного ребру запроса. Вычисление CRPQ запросов сводится поиску всех таких отображений.

3 Вычисление запросов

Рассмотрим основные задачи, возникающие при вычислении конъюнктивных регулярных путевых запросов. Прежде всего следует отметить, что поскольку база данных и запрос представляют собой ориентированные графы, а вычисление запроса требует нахождения отображения вершин запроса в множество вершин базы, задача вычисления запроса аналогична поиску подграфа в графе. Возможность эффективного вычисления запросов связана со следующими вопросами.

Построение плана вычисления запроса. Основными стратегиями вычисления конъюнктивных запросов являются *исчерпывающий поиск* и метод *слияния результатов* вычисления элементарных запросов. Первая стратегия является по сути алгоритмом поиска подграфа. Вторая стратегия предполагает вычисление элементарных запросов и последующее слияние результатов. Как было показано в [11], эффективность применения данных стратегий существенно зависит от структуры вычисляемого запроса. К числу эвристических методов повышения эффективности алгоритма вычисления конъюнктивных запросов можно отнести такие методы как *изменение порядока просмотра вершин запроса*, *«обращение» ребер*, *изменение рорядка обхода исходящих ребер*. В работе [12] предлагаются формальные методы оценки сложности вычисления запроса по заданному плану (на основе перечисленных выше эвристик) и показывается, что применение этих методов повышает скорость вычисления запросов.

Усечение пространства поиска. Если документы базы данных удовлетворяют некоторому набору схем данных, например, графовым схемам [5], то, проверяя запрос на какой-либо одной из них (пусть S), можно отсечь множество документов, на которых поиск по запросу заведомо не даст положительного результата. Проверка документа на схеме представляет собой применение алгоритма, который по запросу Q и графу схемы S позволяет определить факт того, что ни один документ, соответствующий схеме S, не удовлетворяет запросу Q. Если результат такой проверки положительный, то этот факт обозначается как $Q(S) = \emptyset$.

Индексом полуструктурированной базы данных называется дерево $I = \langle \mathcal{S}, T, S_0 \rangle$, где $\mathcal{S}$ — множество графовых схем — вершин иерархии; $T \subseteq (\mathcal{S} \times \mathcal{S})$ — множество ребер иерархии; S_0 — корневая схема иерархии. При этом схемы, являющиеся вершинами иерархии, должны обладать тем свойством, что каждая схема является более общей, чем любая из её дочерних. Далее, для каждого OEM-документа D_i из базы данных должна существовать листовая схема иерархии, которой он соответствует.

Алгоритм усечения пространства поиска представляет собой итерационный процесс. На каждом шаге алгоритма для рассматриваемой вершины S индекса I (для первого шага S, — это корневая схема) проверяем условие $Q(S) = \emptyset$. Если условие выполняется, то «отсекаем» рассматриваемую ветвь индекса, в противном случае переходим к проверке дочерних схем. Продолжаем такие итерации до тех пор, пока не будут рассмотрены все ветви индекса. После обхода дерева получим множество схем (являющихся листьями индекса), для которых $Q(S) \neq \emptyset$. Для остальных листовых схем возьмем все соответствующие им документы и исключим эти документы из пространства поиска.

В работе [13] вводится понятие сложности вычисления запроса по заданному индексу и доказывается, что для любой базы данных существует такой индекс, для которого эта величина принимает наименьшее значение. Применение описанной техники позволяет значительно сократить время вычисления запросов в базах данных, содержащих документы нескольких типов.

Использование материализованных представлений. Материализованные представления, то есть предварительно вычисленные результатов некоторых запросов, широко используются в современных базах данных для уменьшения времени вычисления запросов. Если система получает запрос, результат вычисления которого может быть получен на основании значений материализованных представлений, то часть вычислений, необходимых для получения результата, уже выполнена на этапе построения представлений. Это может приводить к сокращению общего времени вычисления запроса. Основная задача в данном случае состоит в построении перезаписи запроса по системе представлений $\mathbf{V}$, то есть нахождении такого запроса R, что исходный запрос Q представляется в виде $Q(D) = R \circ \mathbf{V}(D)$.

Для модели регулярных путевых запросов проблема выразимости может быть сформулирована следующим образом. Пусть $\varphi : \Delta^* \to \mathrm{Reg}(\Sigma)$ регулярная подстановка (то есть подстановка регулярных языков вместо обозначающих их букв алфавита Δ). Для заданного регулярного языка $Q \in \mathrm{Reg}(\Sigma)$ требуется найти такой язык $R \in \mathrm{Reg}(\Delta)$, что $\varphi(R) = Q$. Однако такое определение приводит к потенциально избыточным перезаписям и в работе [3] предлагается понятие однословой перезаиси, то есть перезаписи, язык которой содержит в точности одно слово, и доказывается, что задача проверки существования такой перезаписи алгоритмически разрешима.

Большое практическое значение имеет задача выбора оптимального набора представлений. Предположим, что нам известна статистика запросов, поступающих к заданной базе данных. Требуется определить такой набор материализованных представлений, который с одной стороны позволяет вычислять эти запросы, а с другой стороны удовлетворяет заданным ограничениям. К числу возможных ограничений относятся ограничения на общее число представлений или ограничения на допустимый объем памяти, который могут занимать построенные представления. В случае регулярных языков существует бесконечно много наборов представлений, которые могут использоваться при вычислении заданного множества запросов. В работе [2] доказывается, что для любой системы запросов можно построить конечное множество представлений, содержащее элементы оптимального базиса.

4 Связь с онтологиями

Распространенным методом формального представления онтологий являются различные дискриптивные логики. Под онтологией в данном случае понимается пара $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, где $\mathcal{T}$ содержит универсальные утверждения, а $\mathcal{A}$ — утверждения об индивидульных объектах.

Дискриптивные логики, использующиеся для описания онтологий, отличаются набором синтаксических конструкций, допустимых при формировании $\mathcal{T}$ и $\mathcal{A}$ утверждений. Рассмотрим в качестве иллюстрации синтаксис логики $\mathcal{ALCOIQ}b^{Self}_{Reg}$ [6]. Пусть $\mathbf{C}, \mathbf{R}, \mathbf{I}$ счетные множества, именуемые множествами *концептов*, *ролей* и *имен обвектов* соответственно. Будем считать, что $\mathbf{C}$ содержит универсальный концепт $\top$ и пустой концепт $\bot$, а также, что $\mathbf{R}$ содержит универсальное отношение $\top$ и пустое отношение B. *Атомарные концепты* B, *концепты* C, *атомарные отношения* R, *простые отношения* S и *отношения*

T подчиняются следующей грамматике, где $a \in \mathbf{I}, A \in \mathbf{C}, P \in \mathbf{R}$ и $P \neq T$:

$$\begin{aligned} B &::= A \mid \{a\} \\ C &::= B \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall T.C \mid \exists T.C \mid \\ &\quad \leqslant nS.C \mid \geqslant nS.C \mid \exists S.Self \\ R &::= P \mid P^- \\ S &::= R \mid S \cap S \mid S \cup S \mid S \setminus S \\ T &::= \top \mid S \mid T \cup T \mid T \circ T \mid T^* \mid id(C) \end{aligned}$$

Множество $\mathcal{A}$ содержит *утверждения* вида $C(a)$, $S(a,b)$ или $a \neq b$, где $C \in \mathbf{C}$, $S \in \mathbf{R}$ и $a, b \in \mathbf{I}$. Множество $\mathcal{T}$ является конечным множеством *аксиом вложения концептов* $C \sqsubseteq C'$ и *аксиом вложения ролей* $S \sqsubseteq S'$, где C и C' концепты, а S и S' — простые роли.

Под *интерпретацией* понимается пара $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, где $\Delta^{\mathcal{I}}$ — непустое множество, именуемое *доменом*, а $\cdot^{\mathcal{I}}$ — интерпретационная функция, сопоставляющая именам объектов элементы домена, концептам — подмножества домена, а отношениям — подмножества декартова произведения $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Выражения, определяющие концепты и отношения, интерпретируются естественным образом, например:

$$(\exists T.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}\ (x,y) \in T^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}.$$

Отношение P^- интерпретируется как обратное по отношению к P. Заметим, что отношения T определяют регулярные языки над алфавитом S.

Интерпретация $\mathcal{I}$ *удовлетворяет* аксиоме вложения концептов $C_1 \sqsubseteq C_2$ (или аксиоме вложения ролей $S_1 \sqsubseteq S_2$), если $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ (или $S_1^{\mathcal{I}} \subseteq S_2^{\mathcal{I}}$). Интерпретация $\mathcal{I}$ является *моделью* онтологии $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, если она удовлетворяет всем утверждениям из $\mathcal{A}$ и аксимомам из $\mathcal{T}$ онтологии $\mathcal{K}$.

Граф интерпретации. Интерпретация онтологии может быть естественным образом представлена в виде ориентированного графа с помеченными ребрами. Пусть задана онтология $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, которая содержит концепт C — «быть женщиной», отношения R и Q, где $R(a,b)$ означает, что a является матерью b, а $Q(a,b)$ означает, что a является сестрой b. Пусть множество $\mathcal{T}$ содержит следующие утверждения:

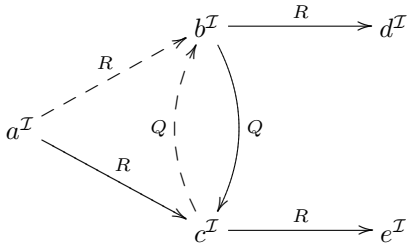
1. $Q \sqsubseteq Q^- \sqsubseteq Q$
2. $R^- \circ R \setminus id(C) \sqsubseteq Q$
3. $R \circ Q \sqsubseteq R$

Первое утверждение говорит о том, что отношение «быть сестрой» является симметричным, второе — что дети одной матери находятся в отношении «быть сестрой», а третье — сестры должны иметь общую мать.

Пусть множество $\mathcal{A}$ содержит следующие утверждения:

$$R(a,c), R(b,d), R(c,e), Q(b,c).$$

Рассмотрим ориентированный граф, в котором вершины соответствуют интерпретациям констант, а ребра, помеченные метками отношений, отражают отношения между ними. Тогда онтологии $\mathcal{K}$ соответствует следующий граф:



Сплошные ребра соответствуют отношениям, полученным на основании утверждений из множества $\mathcal{A}$. Пунктирные ребра были достроены на основании утверждений из $\mathcal{T}$. Исходя из первого утверждения $\mathcal{T}$, получаем, что в множестве $Q^{\mathcal{I}}$ должна содержаться пара $(c^{\mathcal{I}}, b^{\mathcal{I}})$, если интерпретация $\mathcal{I}$ является моделью онтологии $\mathcal{K}$. Аналогично, исходя из

третьего утверждения, получаем, что в множестве $R^{\mathcal{I}}$ должна содержаться пара $(a^{\mathcal{I}}, b^{\mathcal{I}})$. Таким образом представленная на рисунке интерпретация является моделью онтологии $\mathcal{K}$.

Поскольку интерпретация онтологии представляется ориентированным графом, то для поиска могут использоваться описанные ранее языки запросов. Например, CRPQ запрос $xR^*y,\ xR^*z$ возвращает всех родственников (x является общим предком y и z). Следует однако отметить, что онтология может допускать бесконечное число моделей, и вычисление запроса к онтологии следует рассматривать вместе с проблемой построения модели. Описанные методы вычисления запросов необходимо модифицировать для поиска *гарантированных ответов*, то есть ответов, которые справедливы для любой интерпретации данной онтологии.

5 Заключение

В работе описаны основные понятия конъюнктивных регулярных путевых запросов и алгоритмические задачи, возникающие при вычислении таких запросов относительно графовых данных. Показано, что аналогичные задачи возникают при вычислении запросов к онтологиям, заданных с использованием дискриптивной логики. Поскольку заданная онтология может допускать бесконечное число интрпретаций, то задача вычисления запросов сводится к задаче построения гарантированных ответов, то есть ответов, которые справедливы для любой интерпретации данной онтологии. В дальнейшем представляется целесообразным исследовать алгоритмическую сложность этой задачи в зависимости от выразительной силы языка запросов, перезаписи представлений и дискриптивной логики, использующейся для описания онтологии.

Список литературы

[1] Serge Abiteboul and Victor Vianu. Regular path queries with constraints. In *Proc. of the sixteenth ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS 97)*, pages 122–133, 1997.

[2] Sergey Afonin. The view selection problem for regular path queries. In Eduardo Sany Laber and Claudson Bornstein, editors, *Proceedings of the LATIN 2008*, volume 4957 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2008.

[3] Sergey Afonin and Elena Khazova. Membership and finiteness problems for rational sets of regular languages. *International Journal of Foundations of Computer Science*, 17(3):493–506, 2006.

[4] Peter Buneman. Semistructured data. In *PODS'97*, 1997. Invited Tutorial.

[5] Peter Buneman, Susan B. Davidson, Mary F. Fernandez, and Dan Suciu. Adding structure to unstructured data. In Foto N. Afrati and Phokion Kolaitis, editors, *Database Theory—ICDT'97, 6th International Conference*, volume 1186 of *Lecture Notes in Computer Science*, pages 336–350, Delphi, Greece, 8–10 January 1997. Springer.

[6] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. View-based query answering over description logic ontologies. In *Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 242–251, 2008.

[7] James Clark and Steve DeRose. Xml path language (xpath) version 1.0. Technical report, World Wide Web Consortium, 1999.

[8] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. A query language for XML. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1155–1169, 1999.

[9] Alon Halevy, Anand Rajaraman, and Joann Ordille. Data integration: the teenage years. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 9–16. VLDB Endowment, 2006.

[10] D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. Querying semistructured heterogeneous information. In *Deductive and Object-Oriented Databases (DOOD '95)*, number 1013 in LNCS, pages 319–344. Springer, 1995.

[11] С.А. Афонин. Стратегии вычисления регулярных путевых запросов. *Информационные технологии и программирование*, 5(1):9–16, 2002.

[12] С.А. Афонин. Алгоритмы эффективного вычисления конъюнктивных регулярных путевых запросов. *Вычислительные технологии*, 12(2):24–33, 2007.

[13] С. С. Горелов. Оптимальные иерархии схем для поиска по конъюнктивным регулярным путевым запросам в полуструктурированных базах данных. *Программирование*, 4:38–56, 2006.

Работа выполнена при поддержке гранта РФФИ № 09-07-00366-а