УДК 004.822, 519.178

ЗАДАЧИ ОБРАБОТКИ БОЛЬШИХ ГРАФОВ (GRAPH MINING)

Д.И. Черемисинов
Ведущий научный сотрудник лаборатории логического проектирования ОИПИ НАНБ, доцент кафедры инженерной психологии и эргономики БГУИР, кандидат технических наук, доцент

Л.Д. Черемисинова
Главный научный сотрудник лаборатории логического проектирования ОИПИ НАНБ, профессор кафедры инженерной психологии и эргономики БГУИР, доктор технических наук, профессор

Объединенный институт проблем информатики НАНБ, Республика Беларусь
United Institute of Informatics Problems, National Academy of Sciences of Belarus, Republic of Belarus
E-mail: {cher, cld}@newman.bas-net.by

Д.И. Черемисинов
Ведущий научный сотрудник лаборатории логического проектирования ОИПИ НАНБ, доцент кафедры инженерной психологии и эргономики БГУИР, кандидат технических наук, доцент. Область научных интересов: логическое проектирование дискретных устройств управления, разработка моделей, методов и программных средств преобразования структурно-функциональных описаний СБИС.

Л.Д. Черемисинова
Главный научный сотрудник лаборатории логического проектирования ОИПИ НАНБ, профессор кафедры инженерной психологии и эргономики БГУИР, доктор технических наук, профессор. Область научных интересов: логические методы в приложении к вычислительной технике, теория дискретных управляющих устройств, автоматизация логического проектирования.

Аннотация. Графы в настоящее время становятся все более важными для моделирования систем и визуализации информации. В последние годы интеллектуальный анализ графов (graph mining) становится интересной сферой применения для анализа структурированных объектов, таких как химические соединения, белковые структуры, схемы электронных устройств, социальные сети и компьютерные сети. Одним из наиболее важных операций в анализе графов является поиск часто встречающихся подграфов. Основным преимуществом использования часто встречающихся подграфов является ускорение нахождения сходства и сжатого представления (спецификации графа), классификации структур. Обсуждаются общие проблемы для алгоритмов поиска часто встречающихся графов – это представление графа, стратегия поиска, и способ построения.

Ключевые слова: Часто встречающиеся подграфы, интеллектуальный анализ графов, изоморфизм подграфов, КМОП схемы из транзисторов.

Введение. Интеллектуальный анализ данных (data mining) – это извлечение полезной и новой информации из данных. Цель интеллектуального анализа данных – обнаружить полезные алгоритмы, скрытые в наборе данных, и использовать их в качестве явного знания. С увеличением объема и сложности современных данных возникает настоятельная необходимость в ускорении интеллектуального анализа данных большого объема. Естественной формой представления структурированных или частично структурированных данных являются графы. Кроме того, часто данные являются структурными по природе или состоят из

частей и связей между ними. В последнее время анализ графов и управление большими объемами структурированных данных (graph mining) стали важной темой исследований из-за многочисленных приложений в вычислительной биологии, анализе химическом структур, синтезе лекарств и изучении структуры коммуникаций в вычислительных сетях. Традиционные задачи интеллектуального анализа данных (data mining), такие как кластеризация, классификация, поиск часто встречающихся шаблонов и индексация имеют специфику при обработке структурных данных [1-2]. Графы могут представлять широкий спектр данных. Тем не менее, они обычно используются для области исследования, в которой решающее значение имеют отношения. Графы, как общая структура данных, может использоваться для моделирования многих сложных отношений между данными. Метки для вершин и ребер могут представлять различные атрибуты сущностей и отношения между ними.

Области приложения интеллектуального анализа графов характеризуются разными типами графов, поэтому соответствующие проблемы также весьма различаются. Например, графы химических данных относительно малы, и метки на вершинах (обозначающие вид химического элемента) принадлежат небольшому множеству, но могут повторяться много раз в одной молекуле (графе). Это приводит к проблемам, связанным с изоморфизмом графов в приложениях для их анализа. С другой стороны, в приложениях, таких как Интернет, компьютерные сети и социальные сети, метки вершин (например, URL-адреса) различны, и их очень много. Такие графы также являются сложными, потому что распределение степеней этих графов сильно нерегулярно, и это приводит к затруднениям в выборе оптимальной формы хранения. Огромный размер графов компьютерных сетей является серьезной проблемой для алгоритмов интеллектуального анализа графов. Например, в настоящее время граф социально сети Facebook имеет 800 миллионов вершин, а средняя степень каждой вершины равна 130. В 2000 году веб-граф уже имел 2,1 миллиарда вершин и 15 миллиардов ребер. Современные поисковые системы имеют инфраструктуру для поддержки веб-графа с триллионом вершин.

В некоторых случаях графы могут быть изменяющимися во времени. Это означает, что структура графа может быстро меняться. В таких случаях чрезвычайно интересен временной аспект структурного анализа. Примером такого типа являются наборы данных PhoneCall. В этом графе вершинами являются пользователи (индексируемые по номеру телефона), и между двумя пользователями существует ребро, если они разговаривают друг с другом, весом этого ребра служит общая продолжительность вызова. Суммирование длительностей за неделю или месяц даст нам несколько взвешенных графов на одном и том же наборе данных PhoneCall. Набор данных состоит из жителей одного города, использующих одного оператора телефонной связи. Он содержит список всех звонков, сделанных людьми в сети другим пользователям в той же сети за 6 месяцев. У оператора телефонной связи также есть список SMS, отправленных внутри сети (набор SMS). Можно использовать эти наборы для вычисления таких характеристик как распределение длительностей вызовов, обнаружение аномалий, взаимность и т. д.

В последние годы интеллектуальный анализ графов стал популярной областью исследований благодаря его многочисленным приложениям в самых разных практических областях, таких как социология, локализация программных ошибок и компьютерные сети. Анализ графов является важным инструментом для преобразования структурных данных в структурированную информацию. Одним из наиболее важных понятий в интеллектуальном анализе графов является нахождение в нем часто встречающихся подграфов [3], то есть нахождение графов, которые повторяются в основном графе (рис. 1). Нахождение часто встречающихся подграфов (Frequent subgraph mining – FSM) позволяет эффективно структурировать информацию, предлагая метрики, позволяющие охватить различные понятия структуры, например, находить общие структуры белка или общие шаблоны структуры

объектов. Нахождение часто встречающихся подграфов является эффективным инструментом при обнаружении мошенничества путем выявления похожих структур мошеннических транзакций в миллионах электронных платежей.

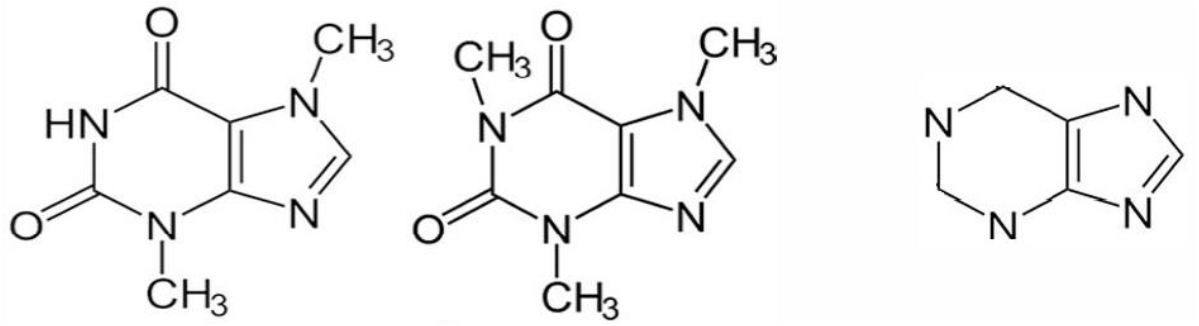

Рисунок 1. Справа налево, структурные формулы теобромина и кофеина, и часто встречающийся подграф этих формул

Поиск часто встречающихся подграфов. Задача FSM состоит в том, чтобы найти все подграфы в заданном наборе данных, частота встречаемости которых превышает указанное пороговое значение. Идея, лежащая в основе FSM, состоит в том, чтобы «наращивать» подграфы-кандидаты либо в ширину, либо в глубину (генерация кандидатов), а затем определять, встречаются ли идентифицированные подграфы-кандидаты достаточно часто в наборе данных графа, чтобы их можно было считать интересными (определение частоты встречаемости). Таким образом, две основные проблемы алгоритма FSM заключаются в том, чтобы эффективно и продуктивно (i) генерировать подграфы-кандидаты и (ii) определять частоту встречаемости сгенерированных подграфов. Эффективная генерация подграфов-кандидатов нужна для предотвращения дублирования кандидатов. Подсчет вхождений требует массового сравнения подграфов-кандидатов с подграфами во входных данных, процесс, известный как проверка изоморфизма.

Существуют две отдельные формулировки задачи FSM: FSM для графовых транзакций и (ii) FSM на основе одного графа. В FSM на основе графовых транзакций входные данные содержат коллекцию графов среднего размера, называемых транзакциями. В FSM на основе одного графа входные данные, как следует из названия, содержат один очень большой граф.

Поиск изоморфного подграфа (subgraph matching). Задача поиска изоморфных подграфов определяется следующим образом [4]: Для графа данных G1 и графа запросов G2 извлекаются все подграфы G, изоморфные G2. Сопоставление подграфа является одной из основных операций решения задачи FSM. Пусть имеются два неориентированных графа G1=(V1, E1) и G2=(V2, E2), где V1, V2 – множество вершин графов, E1, E2 –множество ребер графа. Причем, (|V1|>=|V2|), т.е. G2 является эталонным графом, изоморфный подграф которого ищется в G1. Пусть M – множество (алфавит) возможных меток узлов и ребер графов G1 и G2, а функции m1: V1→M и me_1: E1→M отображают вершины и ребра графа G1 на множество меток; а функции m2: V2→M и me_2:E2→M, соответственно, отображают метки на граф G2. Пусть функция f (биекция) отображает узлы графа G2 на узлы графа G1′=(V1′,E2′), являющегося подграфом G1(f: V2→V1′).

Граф G1′= {V1′, E1′}, являющийся подграфом помеченного графа G1(V1′→V1,E1′→E1), изоморфен графу-эталону G2={V2,E2} в том случае, если существует функция f: V2 →V1′ (биекция), обеспечивающая такое отображение, что

$m2(v2)=m1(f((v2))$ и $me_2(e2)=me_1(fe((e2))$, где $v1 \rightarrow V1'$, $v2 \rightarrow V2$, $e1 \rightarrow E1'$, $e2 \rightarrow E2$. В графе G1 может быть несколько подграфов G1′, изоморфных эталону G2. Тогда частота встречаемости G2| – это число подграфов из G1, изоморфных G2.

Поиск подсхем в КМОП схеме из транзисторов. По аналогии с программированием преобразование иерархической схемы электронного устройства в схему, состоящую исключительно из примитивных элементов, естественно назвать компиляцией. Обратный процесс, в результате которого из схемы строится иерархическая транзисторная схема называется декомпиляцией. Декомпилятор – программа осуществляющая декомпиляцию схем – является одним из инструментов верификации лейаута или перепроектирования (reengineering) схем [2]. Так же как при декомпиляции программ, целью декомпиляции схемы является замена представления схемы на низком (транзисторном) уровне более высокоуровневым ее представлением. В отличие от программного процесса декомпиляция схем не является языковой трансформацией, хотя декомпилятор использует определенный формат (формальный язык представления данных) схем. Языком, используемым декомпилятором схем, является формат SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [5]. Формат SPICE позволяет описывать как схемы транзисторного уровня, так и иерархические.

В формате SPICE электрические схемы состоят из элементов, которые соединены друг с другом цепями (рис. 2), и естественной формальной моделью описания схемы является помеченный неориентированный двудольный граф. Одну долю составляют выводы элементов и порты схемы, а другую – соединения между выводами, т.е. цепи. На рисунках этот граф компактнее представляется в виде его рёберного графа. Для графа G рёберным называется граф L(G), любая вершина которого представляет ребро графа G и две вершины графа L(G) смежны тогда и только тогда, когда соответствующие им рёбра смежны в G. Пометками (раскраской) элементов транзисторного уровня являются названия выводов транзисторов. На рис. 2 в графе КМОП схемы транзисторного уровня у транзисторов не показан вывод подложки, зеленым обозначен вывод «сток», красным – «исток», синим – «затвор». Современные цифровые КМОП схемы содержат до миллиарда примитивных элементов на транзисторном уровне. КМОП (комплементарная структура металл-оксид-полупроводник) – набор полупроводниковых технологий построения интегральных микросхем и соответствующая ей схемотехника микросхем.

Для построения иерархического структурного описания нужно выделить наборы взаимосвязанных транзисторов в схеме в качестве отдельных компонент, т.е. найти подсхемы в исходной плоской схеме (рисунок 2). После замены подсхем из транзисторов элементами описание схемы становится двухуровневым.

Топология двудольных графов, которые моделируют схемы из транзисторов, имеет некоторые специфические свойства. Так как в схеме транзисторного уровня обычно всего два типа элементов: n-МОП и p-МОП транзисторы, то доля выводов элементов двудольного графа кроме выводов самой схемы состоит из вершин степени 1, раскрашенных в 8 цветов соответствии с семантикой выводов этих транзисторов (рис. 2). В доле цепей каждой цепи соответствует компонента связности двудольного графа. Для того чтобы метод Graph Mining был применим, в двудольном графе распределение числа вершин в зависимости от степеней выражалось степенным законом («power law distribution»). Можно считать, что это условие выполняется, так как двудольные графы, которые моделируют схемы из транзисторов, состоят из большого количества небольших локальных цепей, при наличии нескольких цепей с большими степенями.

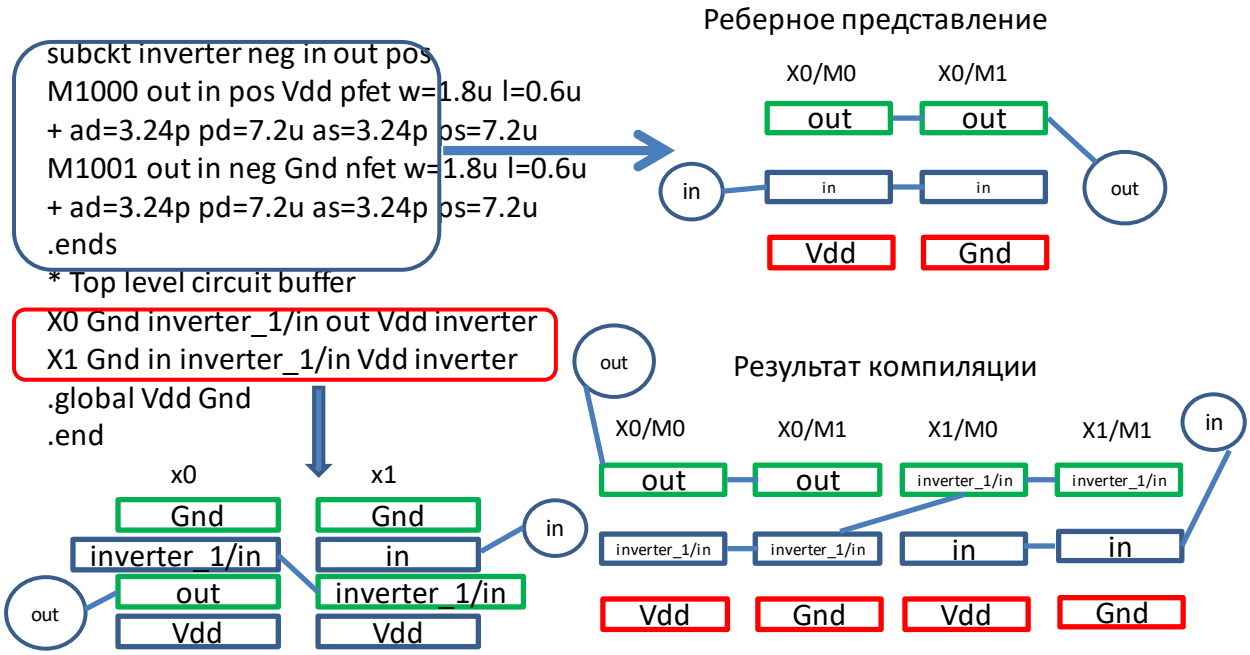

Рисунок 2. Компиляция подсхем (circuit flattening) в формате SPICE

При распознавании подсхемы внутри схемы уровня транзисторов изолируются экземпляры некоторого шаблона (графа, задающего основу подсхемы). Подсхема строится по шаблону включением в граф шаблона вершин выводов подсхемы и ребер, соединяющих эти вершины с цепями шаблона. Для формирования подсхемы цепи шаблона должны быть разбиты на два непересекающихся подмножества: внутренние (локальные) цепи и внешние цепи. В графе подсхемы внешние цепи шаблона соединены с выводами подсхемы. При построении шаблонов разбиением графа на подграфы внешние цепи – это те, что входят в разрез экземпляра шаблона. Для часто встречающихся подграфов внутренние цепи локальны во всех экземплярах подграфа, остальные цепи – внешние цепи шаблона подсхемы.

Заключение. Задача поиска часто встречающихся подграфов (frequent subgraph mining – FSM) стала популярной областью исследований в последнее десятилетие, и к настоящему времени библиография этой задачи насчитывает сотни публикаций. Литература на русском языке по этой проблеме практически отсутствует. За это время было предложено много алгоритмов поиска часто встречающихся подграфов. Трудоемкость FSM огромна из-за необходимости много раз решать задачу изоморфизма подграфа. Алгоритм наивного поиска часто встречающихся подграфов состоит из двух операций. Первая операция разыскивает все подграфы-кандидаты для данного графа G, а вторая подсчитывает частоту встречаемости каждого подграфа-кандидата. Из-за чрезвычайно большой трудоемкости второй операции известные практические алгоритмы FSM ограничиваются поиском подграфов небольшого размера (меньше 10 вершин) – графлетов. Подсчитать оценку частоты встречаемости для графов с миллионами вершин удается для графлетов еще меньшего размера.

Литература

[1]. Koutra D. Kang U, Vreeken J. Faloutsos C. Summarizing and Understanding Large Graphs // Stat. Anal. Data Min. v. 8, n. 3, 2015. – pp.183-202.

[2]. Sun Z., Wang H., Wang H., Shao B., Li J. Efficient subgraph matching on billion node graphs // Proc. VLDB Endow., vol. 5, no. 9, 2012.– pp. 788–799.

[3]. D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years Of Graph Matching In Pattern Recognition // International Journal of Pattern Recognition and Artificial Intelligence, Vol. 18, No. 3 (2004) 265–298.

[4]. Салибекян С.М. Алгоритм поиска изоморфного подграфа в помеченном неориентированном графе

// URL: http://docplayer.ru/29514185-Algoritm-poiska-izomorfnogo-podgrafa-v-pomechennom-neorientirovannom-grafe.html (дата обращения: 18.01.2007).

[5].Baker R.J. CMOS Circuit Design, Layout, and Simulation, Third Edition – Wiley-IEEE Press, 2010.– 1214p.

PROBLEMS OF PROCESSING BIG GRAPHS

D.I. CHEREMISINOV
Ph.Doctor of technical sciences, associate professor
leading researcher

L.D. CHEREMISINOVA
Doctor of technical sciences, professor
principal researcher

United Institute of Informatics Problems, National Academy of Sciences of Belarus, Republic of Belarus
E-mail: {cher, cld}@newman.bas-net.by

Abstract. In recent years, there has been an increased interest in the development of data mining algorithms that work on graphs. Such graphs occur naturally in a number of different areas, such as network attack detection, semantic web, behavioral modeling, re-designing of VLSI, analysis of social networks and classification of chemical compounds. The problem of Graph Mining is to discover typical patterns of graph data. Common feature of such patterns are frequent subgraphs. Graph Mining is one of the arms of data mining in which voluminous complex data are represented in the form of graphs and mining is done to infer knowledge from them. Frequent subgraph mining (FSM) is a sub section of graph mining domain which is extensively used for graph classification, building indices and graph clustering purposes. Identification of frequently occurring graphs / subgraphs in a database or in one large graph is a method that can be used for motif detection, social network monitoring, fraud detection etc.

Electrical circuits consist of elements that are connected to each other by "wires", and the natural formal model of the description of the scheme is a colored undirected bipartite graph. One part consists of the elements' terminals and ports of the circuit, and the other – the connections (nets) between the terminals, i.e. networks (nets) are the "wires". The transformation of the hierarchical circuit of an electronic device into a circuit consisting solely of primitive elements is naturally called a compilation. The reverse process, in the result of which a hierarchical circuit is built up from a flat circuit, is decompilation. To build a hierarchical structural description, you need to select in the circuit as a separate component a set of interconnected transistors, i.e. find subcircuits in the original circuit. After replacing the subcircuits with elements, the description of the circuit becomes two-level. Modern digital circuits contain up to a billion primitive elements at the transistor level. Assuming that subcircuits are frequent subgraphs, you can try to build a decompiler using the Graph Mining methods.

Keywords: Frequent subgraph mining, graph mining, subgraph isomorphism, transistor level CMOS circuits.