УДК: 004.021

# МИНИМИЗАЦИЯ ГРАФОВЫХ МОДЕЛЕЙ АЛГОРИТМИЧЕСКИХ СИСТЕМ

*Чепурко В.А., Грунский И.С.*
*Государственный университет информатики и искусственного интеллекта, г.Донецк*

*Рассматривается задача минимизации ориентированных графов с отмеченными вершинами. Задача минимизации заключается в нахождении разбиения всех вершин графа на классы эквивалентных вершин. Выделены новые классы графов состоящих из одной компоненты сильной связности. Предложены алгоритмы минимизации таких классов графов временной сложности O(e), где e – число ребер графа. Все алгоритмы корректны и выполняют правильное разбиение на классы эквивалентных вершин.*

Графы с отмеченными вершинами являются одной из основных моделей при рассмотрении двух направлений. Первое – это задачи, связанные с анализом операционной среды с помощью блуждающих по ним агентов (мобильных роботов, автоматов, поисковых программ и т.п.) относятся к категории традиционных задач искусственного интеллекта. Одной из центральных и актуальных как в теоретическом, так и в прикладном аспектах проблем, возникающих при исследованиях взаимодействия автоматов и операционной среды, является проблема анализа или распознавания свойств этой среды при различной априорной информации и при различных способах взаимодействия автомата и операционной среды. В таких задачах рассматриваются различные геометрические модели сред.

Операционная среда рассматривается как ориентированный граф с помеченными вершинами[1]. Такие графы возникли первоначально как блок-схемы и схемы программ, а в настоящее время находят применение в задачах навигации роботов[2].

Второе направление – это блок-схемы программ. Задача эквивалентности программ относится к числу наиболее важных задач теории программирования. Особое значение задачи эквивалентности обусловлено тем, что именно к этой проблеме эквивалентных преобразований сводится большинство задач оптимизации, верификации и анализа программ[2]. Поэтому разработка и внедрение эффективных алгоритмов проверки эквивалентности программ оказывает заметное влияние на повышение производительности и качества многих инструментальных средств анализа и преобразования программ. В последние годы наряду с задачами повышения эффективности и надёжности программ не меньшую актуальность приобрела задача своевременного обнаружения постороннего кода, которая включает выявление недекларированных возможностей программных продуктов, а также обнаружение и устранение программ-вирусов.

В обоих случаях такие графы могут содержать большое количество вершин, поэтому возникает задача уменьшения их количества с сохранением всех графа. Эта задача известна как задача минимизации.

В настоящей работе рассматривается проблема минимизации ориентированных ациклических графов с отмеченными вершинами. Эта проблема позволяет находить эквивалентные графы, с помощью которых можно представить различные системы. Анализ графов проводится методами, аналогичными методам теории автоматов. Эти методы созданы для графовых систем, не являющихся конечными автоматами, но являющимися в некотором смысле автоматоподобными системами.



Проблема минимизации состоит в нахождении разбиения всех вершин графа на классы эквивалентных. В [1] предложен алгоритм минимизации графа временной сложности $O(2^n)$ в общем случае и $O(n^2)$ для так называемых детерминированных[2] графов, где n – число вершин в графе.

При минимизации на временную сложность алгоритмов сильно влияет структура графа, так для деревьев и ациклических графов предложены алгоритмы временной сложности O(e), где e – число дуг графа. В данной работе был проведен анализ графов состоящих из одной компоненты сильной связности и выделено несколько видов графов, для которых временная сложность минимизации существенно уменьшается.

**Постановка задачи**

Пусть G=(V,E,M,μ) конечный, помеченный, ориентированный граф, без петель и кратных дуг[1], где V – множество вершин, $E \subseteq V\times V$ множество дуг, $E(v)=\{u\in V|(v,u) \in E\}$ – множество последующих вершин и $E^{-1}(v)=\{u\in V|(u,v) \in E\}$ – множество предшествующих вершин, M – множество номеров меток, $\mu$: $S \rightarrow M$ – функция размечивания вершин. Конечную последовательность вершин $p=g_1\dots g_k$ такую, что $g_{i+1} \in E(g_i)$, $1<i<=k$, назовём путём в графе G. Слово $\mu(p)=\mu(g_1)\dots\mu(g_k)$ назовём отметкой пути p. Отметку любого пути, исходящего из вершины $g \in V$, будем называть словом, порожденным вершиной g. Язык Lg определим как множество всех слов, порождённых вершиной g. Вершины g, h – назовём эквивалентными, если Lg=Lh, и отличимыми в противном случае. Граф G называется приведенным, если все его вершины попарно отличимы. Число вершин в E(v) назовём степенью вершины v. Через $\mu[E(v)]$ обозначим множество всех меток вершин из E(v). Через |V| обозначим мощность множества V. Назовём граф G ранга t, где t – наименьшее число рёбер, которые нужно удалить из графа для того, что бы он стал ацикличным. Считаем что G состоит из одной компоненты сильной связности. Граф, в котором для всех $v_i\in V$ $|E(v_i)|=1$ назовём простым циклом. Вершины $v_j\dots v_k$ образующие собой ациклический подграф графа G назовём линзой, в которой вершина $v_j$ – начало линзы, у которой $|E(v_j)|>1$, а $v_k$ – конец линзы с $|E^{-1}(v_k)|>1$. При удалении одного ребра соединяющего две любые вершины линзы существует путь, соединяющий начальную и конечную вершины линзы. А так же при удалении одной из вершин линзы, кроме начальной и конечной, существует хотя бы 1 путь из начальной вершины в конечную.

8

В данной работе проведён анализ графов состоящих из одной компоненты сильной связности ранга 1 и выделено несколько видов графов, временная сложность минимизации которых существенно уменьшается по сравнению с общим алгоритмом.

Первый тип графов простой цикл. Рассмотрим метод минимизации простых циклов. Берём произвольную вершину цикла v, и последовательно просматриваем вершины цикла, используя преемника E(v) обрабатываемой вершины v. Осуществляем обход всех его вершин, при этом проверяем, является ли слово отметок периодичным. Если да, то кратчайшему периоду слова отметок вершин соответствует минимальный граф. В противном случае, если повторяющейся подпоследовательности отметок нет, все вершины попарно отличимы. Определена временная сложность алгоритма для ориентированных циклов с отмеченными вершинами равная O(n), где n – число вершин.

На рис. 1 показан граф G1, в котором повторяются отметки 1, 2, таким образом, минимальный граф состоит из 2-х вершин.

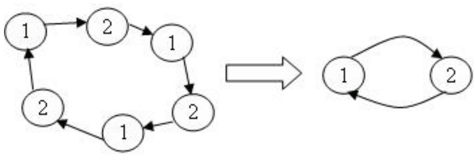

Рисунок 1 – Граф G1

На рис. 2 изображён граф G2. Цепочка вершин цикла не содержит повторяющихся последовательностей, поэтому граф есть минимальным.

Следующий тип графов – это простой цикл с одной линзой. Метод минимизации такого графа заключается в следующем: берём



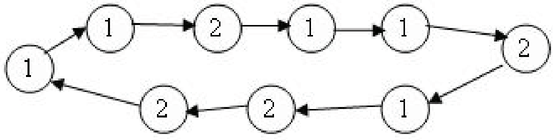

Рисунок 2 – Граф G2

вершину графа $v_k$, которая является концом линзы и начиная с данной вершины, выполняем минимизацию алгоритмом для ациклических графов[3] пока не встретим начальную вершину линзы. Остальные вершины цикла помещаются все в разные классы эквивалентности. Определена временная сложность алгоритма для ориентированных циклов с отмеченными вершинами равная O(n+p), где p – число рёбер линзы, а n – число вершин графа G.

На рис. 3 показан граф G3 и его минимальный граф. Выбираем конечную вершину линзы с номером 7. Далее смотрим предшественников $E^{-1}(7)=\{5, 6\}$ для этой вершины, так как метки одинаковы, то вершины 5, 6 эквивалентны.

Рассмотрим метод минимизации простых циклов с несколькими линзами. Он заключается в следующем: берём

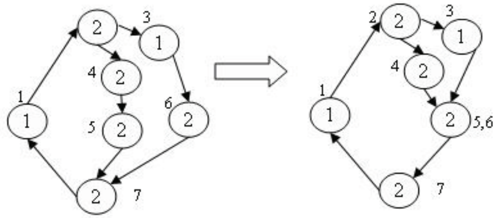


Рисунок 3 – Граф G3

вершину графа $v_k$, которая является концом одной из линз и начиная с данной вершины, выполняем минимизацию алгоритмом для ациклических графов пока не встретим первую обработанную вершину. Далее обходим вершины и линзы графа и ищем среди них повторяющуюся подпоследовательность. Определена временная сложность алгоритма для ориентированных циклов с отмеченными вершинами равная O(e), где e – число рёбер графа.

На рис. 4 показан граф G4 и этапы его минимизации. При первой обработке графа начиная с 1 вершины найдены 2 эквивалентные вершины {5,6} при этом изменилась конечная вершина первой линзы графа G4 и стала вершина {5,6} вместо 7 вершины. Рассматриваемый граф состоит из двух линз {1,7,8,9} и {2,3,4,5,6}. При проверки повторяющегося периода выяснилось что

8

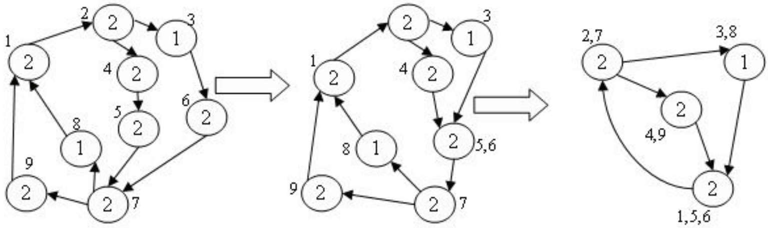

Рисунок 4 – Граф G4

линзы одинаковые, следовательно последовательность двух линз превращается в одну линзу.

Показано, что алгоритмы общего вида для данных классов графов избыточны, т.к. их сложность O(m*n*log n). В работе предложены новые алгоритмы минимизации частных классов графов, простых циклов, простых циклов с линзой и простых циклов с несколькими линзами сложности O(e), в которых существенно используется структура графа. В данных алгоритмах не требуется начальное разбиение вершин графа. Доказано, что результатом работы алгоритма являются классы эквивалентных вершин, то есть он решает задачу минимизации.

**Литература**

[1] Сапунов С.В. Анализ графов с помеченными вершинами: Дисс. канд. физ.-мат. наук: 27.09.07. – Донецк.: 2007. – 150 с.

[2] Dudek J., Jenkin M., Milios E., Wilkes D. Map validation and Robot Self-Location in a Graph-Like World // Robotics and Autonomous Systems, 1997.-V.22(2). – P. 159-178.

[3] Чепурко В.А. Минимизация ориентированных ациклических графов с отмеченными вершинами: материалы VI межд. научно-практ. конф. «Мат. и прогр. обеспечение интеллектуальных систем». (Днепропетровск, 12-14 ноября 2008 г.) / Днепропетровск национальный университет, 2008. – С. 331-332

8