

Московский государственный университет
имени М. В. Ломоносова

На правах рукописи

Русаков Дмитрий Михайлович

СХЕМЫ ПРОГРАММ С КОНСТАНТАМИ

Специальность 01.01.09 – дискретная математика и
математическая кибернетика

АВТОРЕФЕРАТ

диссертации на соискание учёной степени
кандидата физико-математических наук

Москва – 2008

Работа выполнена на кафедре математической кибернетики факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова.

Научный руководитель:	доктор физико-математических наук, профессор Подловченко Римма Ивановна.
Официальные оппоненты:	доктор физико-математических наук, профессор Ломазова Ирина Александровна, кандидат физико-математических наук, доцент Хачатрян Владимир Ервандович.
Ведущая организация:	Институт системного программирования Российской Академии наук.

Защита состоится 12 декабря 2008 г. в 11.00 на заседании диссертационного совета Д 501.001.44 при Московском государственном университете имени М. В. Ломоносова по адресу: 119991, ГСП-1, Москва, Ленинские горы, МГУ, 2-й учебный корпус, факультет ВМиК, аудитория 685.

С диссертацией можно ознакомиться в библиотеке факультета ВМиК МГУ.

С текстом автореферата можно ознакомиться на официальном сайте факультета ВМиК Московского государственного университета имени М. В. Ломоносова <http://www.cs.msu.su> в разделе «Наука» – «Работа диссертационных советов» – «Д 501.001.44».

Автореферат разослан __ ноября 2008 г.

Учёный секретарь
диссертационного совета
профессор



Трифонов Н. П.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Диссертационная работа относится к теории алгебраических моделей программ и посвящена изучению семантических свойств моделей, именуемых моделями программ с константами.

В теории программирования изначально модели программ создавались для исследования различных свойств самих программ, и главным из них является отношение функциональной эквивалентности программ. Алгебраические модели программ введены Р.И. Подловченко в 1981 году как обобщение ранее известных моделей, объектами которых являются операторные схемы Ляпунова-Янова, и моделей, называемых дискретными преобразователями Глушкова-Летичевского. Объекты алгебраических моделей программ называются схемами программ, и отдельная модель определяется введением отношения эквивалентности в множестве схем программ.

Предпринятое обобщение базируется на следующих концепциях:

- отталкиваться от предварительно заданной формализации понятия программы;
- при переходе от формализованной программы к моделирующей её схеме сохранять структуру самой программы.

Преследуемая при этом цель состоит в том, чтобы рассматривать модели программ, обладающие свойством: для них существуют аппроксимируемые ими классы формализованных программ, то есть такие классы, в которых функциональная эквивалентность программ следует из эквивалентности построенных для них схем, принадлежащих модели.

Изложим вкратце, как реализован этот замысел.

Формализованные программы строятся над базисом операторов и логических условий применением всех традиционных средств композиции операторов, кроме аппарата процедур. Эти средства реализуются в управляющем графе программы, вершины которого нагружены базисными операторами и логическими условиями.

Переход от программы к моделирующей её схеме осуществляется заменой операторов их символами, а логических условий — булевыми переменными. При этом управляющий граф программы сохраняется в её схеме. Множество операторных символов и булевых переменных составляет базис, над которым строятся схемы программ.

Чтобы ввести отношение эквивалентности в множестве схем программ над заданным базисом, описывается процедура выполнения схемы в ситуации, когда на любой цепочке операторных символов заданы значения всех булевых переменных. Само отношение эквивалентности определяется двумя параметрами: множеством допустимых ситуаций, в которых выполняются схемы, и решением, какие операторные цепочки, воспринимаемые процедурой выполнения как результаты его, считать эквивалентными. При заданных параметрах, по определению, две схемы полагаются эквивалентными, если, какой бы ни была допустимая ситуация, в которой они выполняются, всякий раз, как выполнение одной из них завершаемо, завершаемо и выполнение другой, и при этом выполненные операторные цепочки эквивалентны. Вводится и отношение включения одной схемы в другую, при котором изложенные выше требования выполняются только для первой схемы.

Сообразуясь с общей целью построения алгебраических моделей программ, центральной в проблематике их теории является проблема эквивалентности в модели. Она рассматривается в множестве всех схем программ, принадлежащих выбранной модели, и состоит в поиске алгоритма, распознающего, эквивалентны или нет две произвольные схемы из этого множества. Подобным образом формулируется и проблема включения в модели, когда распознаванию подлежит отношение включения. Очевидно, что из разрешимости в модели проблемы включения следует разрешимость в ней проблемы эквивалентности. Именно это и сообщает интерес к проблеме включения.

В теории алгебраических моделей программ найден достаточный признак того, что модель является аппроксимирующей,

и изучаются только аппроксимирующие модели. Установлено, что среди них имеются как модели с разрешимой проблемой эквивалентности, так и модели с неразрешимой проблемой эквивалентности. При обращении к проблеме включения обнаружен факт существования моделей с неразрешимой проблемой включения, но с разрешимой проблемой эквивалентности.

Среди аппроксимирующих моделей программ особенно активно изучаются так называемые полугрупповые модели. Индуцирующая их эквивалентность операторных цепочек обладает свойством: классы эквивалентных цепочек образуют полугруппу по операции конкатенации цепочек. Множество допустимых ситуаций, в которых выполняются схемы, полностью определяется отношением эквивалентности операторных цепочек.

Для многих полугрупповых моделей, удовлетворяющих требованию: определяющие их полугруппы не обладают циклами, найдены приемлемые по сложности алгоритмы распознавания в них проблемы эквивалентности. Подверглись рассмотрению и полугрупповые модели, для которых полугруппа имеет циклы. Они имеют следующий тип: среди базисных операторных символов имеются символы, названные константами, и две операторные цепочки полагаются эквивалентными в одном из двух случаев: либо обе не содержат входящих констант и полностью совпадают, либо обе имеют входящие константы, и тогда совпадают хвосты цепочек, начинающиеся последним входящим константы. Такие модели названы нами моделями программ с константами. Параметром отдельной модели является число используемых ею констант.

Описанные модели аппроксимируют классы программ, обладающих свойством: при выполнении программы встреча с оператором, соответствующим константе, устанавливает текущее состояние памяти, каким бы оно ни было, в фиксированное состояние, определяемое этой константой, то есть фактически встреча с оператором-константой аннулирует предшествующую работу программы.

Модели программ с константами изучались А.А. Летичев-

ским. Им установлена разрешимость проблемы эквивалентности в этих моделях, однако факт существования разрешающего алгоритма не сопровождается его построением. Вместе с тем ощущается сложность распознавания эквивалентности, ставящая под вопрос практическое использование процедуры распознавания.

Факт разрешимости проблемы эквивалентности в моделях программ с константами можно извлечь из результатов исследований моделей вычислений, введённых Л.П. Лисовиком. Однако этот окольный путь не применялся для оценки сложности распознавания эквивалентности.

Проблема включения вообще не рассматривалась для моделей программ с константами.

Актуальность исследований. Из сделанного нами очерка исследований в теории алгебраических моделей программ вытекает актуальность выявления места, занимаемого моделями программ с константами среди полугрупповых моделей, для которых установлена разрешимость проблемы эквивалентности в модели. Это место определяется оценкой ёмкостной сложности разрешения проблемы эквивалентности в моделях программ с константами.

Цель исследований. Цель исследований состоит в следующем: установить разрешимость проблемы включения в произвольной модели программ с константами; найти оценки ёмкостной сложности разрешения проблемы включения, в результате чего будут получены оценки ёмкостной сложности разрешения проблемы эквивалентности.

Основные результаты работы и научная новизна. В диссертации исследованы проблемы включения и эквивалентности в алгебраической модели программ с константами.

Основные результаты данной диссертации следующие:

1. Построен алгоритм разрешения проблемы эквивалентности в множестве схем программ, принадлежащих произвольной алгебраической модели программ с константами и

имеющих не более одного вхождения любой константы в схему.

2. Впервые рассмотрена проблема включения в произвольной алгебраической модели программ с константами и доказана её разрешимость. Установлено, что проблема включения, следовательно, и проблема эквивалентности, в этой модели являются PSPACE-полными.
3. Построен альтернативный алгоритм, разрешающий проблему включения в алгебраической модели программ с константами. Дана оценка его ёмкостной сложности, и этим установлена оценка ёмкостной сложности алгоритма, решающего проблему эквивалентности.

В итоге выявлен статус произвольной алгебраической модели программ с константами среди полугрупповых моделей программ с установленной для них разрешимостью проблемы эквивалентности. Применимость в практике программирования алгоритма, распознающего эквивалентность в алгебраической модели программ с константами, находится на грани возможного.

Теоретическая и практическая ценность. Работа носит теоретический характер. Предложенные в ней методы и алгоритмы могут быть использованы при решении задач верификации, оптимизации, синтеза и реорганизации программ, аппроксимируемых моделями программ с константами.

Апробация работы. Представленные в диссертации результаты докладывались на семинаре факультета ВМиК МГУ по теоретическим вопросам программирования под руководством профессора Р.И. Подловченко и доцента В.А. Захарова, на V-й международной конференции “Дискретные модели в теории управляющих систем” (2003, Москва), на VI-й международной конференции “Дискретные модели в теории управляющих систем” (2004, Москва), на всероссийской конференции студентов, аспирантов и молодых учёных “Технологии Microsoft в теории и практике программирования”, (2005, Москва), на VII-й меж-

дународной конференции “Дискретные модели в теории управляющих систем”, (2006, Москва).

Структура и объем работы. Диссертация состоит из введения, пяти глав, заключения и списка литературы, включающего 37 наименований. Общий объем диссертации составляет 90 страниц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Диссертация имеет следующую структуру. Она состоит из введения, пяти глав и краткого заключения.

Во введении описывается область исследований, к которой относится данная работа, и рассматриваемые в ней проблемы.

Первая глава диссертации содержит формальное определение алгебраической модели программ с константами и проблем включения и эквивалентности в ней.

Здесь же используется известный ранее результат о сводимости проблем включения и эквивалентности в рассматриваемых моделях к одноимённым проблемам в классе унифицированных схем, принадлежащих модели (лемма 1). Унификация схемы состоит в группировке в одинакового вида фрагменты разбросанных ранее по схеме логических условий.

Далее теоремой 1 устанавливается сводимость проблем включения и эквивалентности в множестве унифицированных схем к одноимённым проблемам в множестве матрично-алгебраических схем программ. Алгоритм сведения устанавливает взаимно-однозначное соответствие между элементами этих множеств.

Пусть \mathcal{M}_∞ — изучаемая алгебраическая модель программ с константами.

Во второй главе ставится задача: построить алгоритм, разрешающий проблему включения в множестве \mathcal{M}_1 схем программ, принадлежащих модели \mathcal{M}_∞ и таких, в которые любая константа из заданного базиса входит в схему не более одного раза.

Цель решения этой задачи — прощупать методику разрешения проблемы включения в общем случае, то есть для \mathcal{M}_∞ .

Методика получения этого результата следующая. Вначале устанавливается, что для проверки включения схем в \mathcal{M}_1 достаточно научиться проверять включение в множестве схем программ без констант и проверять включение областей определения схем в \mathcal{M}_1 (теорема 2). Областью определения схемы называется множество ситуаций, на которых выполнение схемы завершается.

Далее выявляется, что для проверки включения областей це-

лесообразно построить специальные объекты, именуемые нами диаграммами. В диаграммах учтено следующее обстоятельство: выполнение схемы из \mathcal{M}_∞ незавершаемо, если путь его проходит более одного раза через вершину с константой (утверждение 2). Диаграмма схемы представляет собой дерево, вершины которого помечены константами; при этом метки вершин одной ветви диаграммы не повторяются. Дуги диаграммы помечены схемами программ без констант. Описан алгоритм построения по унифицированной схеме из \mathcal{M}_1 соответствующей ей диаграммы (лемма 3).

Теоремой 3 устанавливается сведение проблем включения и эквивалентности в множестве схем из \mathcal{M}_1 к проблеме включения в множестве диаграмм, соответствующих схемам из \mathcal{M}_1 . В частности, проблема включения областей определения схем из \mathcal{M}_1 сводится к проблеме включения областей определения диаграмм. Последняя проблема, в свою очередь, сводится к проблеме включения областей определения схем программ без констант (лемма 4). Решение этой проблемы сводится к проверке определённых отношений в множестве конечных автоматов, которые строятся по схемам без констант (теорема 4).

Третья глава посвящена выявлению сложности проблемы включения в \mathcal{M}_∞ . Теоремой 6 доказано, что эта проблема является PSPACE-полной. Параметром, по которому оценивается сложность распознавания отношения включения одной схемы в другую, является общее число вершин с константами в обеих сравниваемых схемах.

Для доказательства теоремы 6 строится алгоритм, разрешающий проблему включения в модели и использующий для этого не более чем полиномиальную память (лемма 6). Этот алгоритм для каждой пары ветвей диаграмм сравниваемых схем использует три конечных автомата. Эти автоматы имитируют параллельное выполнение схем программ без констант, построенных для каждой дуги ветвей диаграмм; при этом учитываются константы, соответствующие вершинам ветвей. Задача алгоритма состоит в проверке пустоты языков этих автоматов. Про-

странства состояний этих автоматов экспоненциальны, поэтому алгоритм не выполняет их полного построения в памяти. Разработанный алгоритм позволяет проверить пустоту автоматов, строя их фрагменты “на лету”.

Таким образом устанавливается, что проблема входит в класс PSPACE.

PSPACE-полнота доказывается сведением известной PSPACE-полной проблемы — проблемы пересечения языков k конечных автоматов — к проблеме включения схем в \mathcal{M}_∞ .

Итак, в третьей главе диссертации предложен алгоритм, решающий проблему включения в \mathcal{M}_∞ и требующий для своей работы полиномиальной памяти. Этот алгоритм не применим в практическом программировании, причиной чего является “трюк”, использованный при проверке пустоты языков автоматов, пространство состояний которых экспоненциально.

В связи с этим четвертая глава посвящена построению альтернативного алгоритма, решающего проблему включения в \mathcal{M}_∞ и применимого на практике. Идеи данного алгоритма развивают идеи алгоритма проверки включения в \mathcal{M}_1 . Теперь их реализация становится значительно сложнее. Так, если проблема включения в \mathcal{M}_1 сводилась к проверкам отношений

$$G_1 \sqsubseteq G_2$$

и

$$Dom(G_1) \subseteq Dom(G_2),$$

то проблема включения в \mathcal{M}_∞ сводится к проверке отношений

$$G_1 \stackrel{\mathcal{L}_0}{\sqsubseteq} G_2, \text{ где } \mathcal{L}_0 = Dom(G_3) \cap Dom(G_4) \cap \dots \cap Dom(G_n),$$

и

$$Dom(G_1) \cap Dom(G_2) \cap \dots \cap Dom(G_{n-1}) \subseteq Dom(G_n).$$

Здесь G_1, G_2, \dots, G_n — схемы программ без констант; $Dom(G)$ — область определения схемы G ; а отношение $G_1 \stackrel{\mathcal{L}_0}{\sqsubseteq} G_2$ справедливо в том и только том случае, когда выполняется требование: если схема G_1 завершит работу на ситуациях из множества \mathcal{L}_0 , то на этих ситуациях завершит работу и схема G_2 , и полученные ими результаты совпадают.

В четвёртой главе дано крупноблочное описание алгоритма, распознающего включение в \mathcal{M}_∞ , и доказана его корректность (теорема 8).

В пятой главе детально выписывается сам алгоритм и строится оценка его сложности по используемой памяти (теорема 9); она выглядит следующим образом:

$$O(n^{6k}),$$

где n — общее число вершин в сравниваемых схемах, а k — общее число вершин с константами в сравниваемых схемах.

В итоге проведёнными исследованиями моделей программ с константами выявлен её, в известном смысле, пограничный статус среди полугрупповых моделей программ с установленной для них разрешимостью проблемы эквивалентности. Применимость на практике алгоритма распознавания эквивалентности схем программ с константами на грани возможного.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

В диссертации исследованы проблемы включения и эквивалентности в алгебраической модели программ с константами.

Основные результаты данной диссертации следующие:

1. Построен алгоритм разрешения проблемы эквивалентности в множестве схем программ, принадлежащих произвольной алгебраической модели программ с константами и имеющих не более одного вхождения любой константы в схему.
2. Впервые рассмотрена проблема включения в произвольной алгебраической модели программ с константами и доказана её разрешимость. Установлено, что проблема включения, следовательно, и проблема эквивалентности, в этой модели являются PSPACE-полными.
3. Построен альтернативный алгоритм, разрешающий проблеме включения в алгебраической модели программ с константами. Дана оценка его ёмкостной сложности, и этим установлена оценка ёмкостной сложности алгоритма, решающего проблему эквивалентности.

В итоге выявлен статус произвольной алгебраической модели программ с константами среди полугрупповых моделей программ с установленной для них разрешимостью проблемы эквивалентности. Применимость в практике программирования алгоритма, распознающего эквивалентность в алгебраической модели программ с константами, находится на грани возможного.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

- [1] Подловченко Р. И., Русаков Д. М. Каноническая форма схемы программ с однократным вхождением константы // Труды V международной конференции “Дискретные модели в теории управляющих систем” (Ратмино 26-29 мая 2003).— М.: изд. отдел ф-та ВМиК МГУ, 2003.— С. 65–66.
- [2] Подловченко Р. И., Русаков Д. М. О проблеме включения для схем программ с константами // Труды VI международной конференции “Дискретные модели в теории управляющих систем” (Москва, 7-11 декабря 2004).— М.: изд. отдел ф-та ВМиК МГУ, 2004.— С. 137–140.
- [3] Подловченко Р. И., Русаков Д. М. Схемы программ с константами // Программирование.— 2005.— № 3.— С. 5–18.
- [4] Подловченко Р. И., Русаков Д. М. Проблема включения в алгебраической модели программ с константами // Программирование.— 2007.— № 3.— С. 3–15.
- [5] Русаков Д. М. О проблеме эквивалентности схем программ с константами // Сборник тезисов лучших дипломных работ 2004 года.— М.: изд. отдел ф-та ВМиК МГУ, 2004.— С. 61–63.
- [6] Русаков Д. М. О проблеме включения для схем программ с константами // Технологии Microsoft в теории и практике программирования.— 2005.— С. 134.
- [7] Русаков Д. М. Проблема эквивалентности схем программ с константами // Труды VII международной конференции “Дискретные модели в теории управляющих систем” (Петровское, 4-6 марта 2006 г.).— М.: МАКС Пресс, 2006.— С. 309–314.
- [8] Русаков Д. М. Алгоритм проверки включения схем в алгебраической модели программ с константами // Программирование.— 2007.— № 5.— С. 3–13.
- [9] Podlovchenko R., Rusakov D., Zakharov V. The equivalence problem for programs with mode switching is PSPACEcomplete

// В сборнике “Труды института системного программирования”.— М.: Институт Системного Программирования РАН, 2006.— Т. 11.— С. 109–128.

- [10] Podlovchenko R., Rusakov D., Zakharov V. On the equivalence problem for programs with mode switching // Lecture Notes in Computer Science.— 2006.— Vol. 3845.— Pp. 351–352.