Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики
и кибернетики

ТРУДЫ

V Международной конференции
"Дискретные модели в теории управляющих систем"

(Ратмино, 26 – 29 мая 2003 года)

Москва, 2003

УДК 510.5, 519.71
ББК 22.12:22.18
Т78

Труды V Международной конференции "Дискретные модели в теории в теории управляющих систем" (Ратмино, 26 – 29 мая 2003 г.) – М.: Издательский отдел факультета ВМиК МГУ имени М. В. Ломоносова (лицензия ИД N 05899 от 24.09.2001), 2003 – 100 с.

Сборник содержит тезисы докладов, представленных на V Международную конференцию "Дискретные модели в теории управляющих систем". Тематика конференции охватывает такие разделы математической кибернетики, как комбинаторика, теория функциональных систем, теория кодирования и криптография, синтез управляющих систем, теория автоматов, сложность алгоритмов, математическая логика и теория алгоритмов, теория программирования.

Конференция организована кафедрой математической кибернетики факультета ВМиК МГУ имени М. В. Ломоносова.

ISBN 5-89407-162-3

Proceeding of the V International Conference on Discrete Models in Control System Theory (Ratmino, May 26 – 29, 2003), – Moscow: Publishing Department of Faculty of Computational Mathematics and Cybernetics of Lomonosov's MSU (licence ID N 05899 of 24.09.2001), 2003. – 100 p.

This book contains the thesises submitted to the V International Conference on Discrete Models in Control System Theory. This conference covers such branches of mathematical cybernetics as combinatorics, theory of functional systems, coding theory and cryptography, control system synthesis, computational complexity, mathematical logic and algorithm theory, theory of programing.

The conference is organized by Chair of Mathematical Cybernetics of Faculty of Computational Mathematics and Cybernetics of Moscow State University named by M. V. Lomonosov.

ISBN 5-89407-162-3

Содержание

Аблаев Ф. М. Сложность квантовых моделей вычислений 6
Аблаев Ф. М., Гайнутдинова А. Ф. Сложность классического моделирования квантовых машин 6
Аксенова Е. А., Лазутина А. А., Соколов А. В., Тарасюк А. В. Об оптимальном управлении динамическими структурами данных 7
Алексеев В. Б. Мощность максимального независимого множества для одного предиката на E_3 9
Алексеев В. Б., Вороненко А. А., Селезнева С. Н. О сложности реализации функций k-значной логики поляризованными полиномами 10
Алехина М. А. О надежности схем в базисе $\{\vee, \bar{}\}$ при неисправностях типа 0 на выходах элементов 11
Андреева Т. В. О логарифмической выпуклости мощностей слоев k–значной n–мерной решетки 13
Бондаренко Л. Н. Многочлены композиций и обобщенные многочлены Эйлера 15
Борисов А. Е. Закономерности в деревьях вывода для стохастической разложимой КС-грамматики 17
Варновский Н. П., Захаров В. А. К вопросу о существовании стойких обфускаторов программ 19
Викторова М. С., Захаров В. А. Об одной системе вывода, связанной со статическим анализом программ 21
Вознесенская Т. В., Костенко В. А., Маркин М. И. Задача и алгоритм календарного планирования работ сервисного подразделения IT-компании 23
Воронин В. П., Трифонова Т. В. Некоторые весовые и метрические свойства кодов Хэмминга 24
Грибок С. В. О нижних мощностных оценках функций Шеннона 27
Грунская В. И. Сохранение топологических свойств среды при ее преобразованиях 29
Грунский И. С., Сапунов С. В. О контроле детерминированных графов . 29
Жильцова Л. П. Об одной системе линейных уравнений для стохастического КС-языка 31
Захарьящев И. М., Захаров В. А. Об одной полисемантической модели последовательных программ 35
Иванчик И. И. Приложение двойных неупорядоченных разбиений к решению одного нелинейного дифференциального уравнения второго порядка 36
Иорданский М. А. Базисы планарных графов 38
Иоссель М. А., Рублев В. С. Язык отчетов для динамической информационной модели DIM 40
Калмыков Г. И. Каркасная классификация помеченных блоков 41
Карповский А. В., Рублев В. С. Оптимизация производственного плана по прибыли 44

Костенко В. А. Проблемы разработки и применения итерационных алгоритмов для построения расписаний 46
Козловский В. А., Толмачевская Л. А. О восстановлении автоматов по траекториям в пространстве состояний 46
Кузюрин Н. Н. Вероятностный анализ жадного алгоритма для задачи о покрытии 48
Левенштейн В. И. Корректирующая способность двоичных линейных кодов и монотонные функции 49
Лобачев Д. И., Рублев В. С. Взаимодействия динамической информационной модели DIM 51
Ложкин С. А. О структуре минимальных схем из функциональных элементов в базисе $\{\&, \vee, \neg\}$, реализующих линейную функцию 52
Мерекин Ю. В. Суффиксный метод получения нижних оценок сложности схем композиции слов 53
Михеева Е. А. К вопросу описания решетки замкнутых классов конечнозначной логики 54
Нагорный А. С. Сложность сортировки k-значного n-мерного куба 56
Омельянов К. Г., Сапоженко А. А. О числе и структуре множеств, свободных от сумм в отрезке натуральных чисел 57
Охотин А. С. Системы языковых уравнений и замкнутые классы функций алгебры логики 58
Подловченко Р. И., Долгих Б. А. Двухступенчатое моделирование программ с процедурами 66
Подловченко Р. И., Русаков Д. М. Каноническая форма схемы программ с однократным вхождением константы 68
Подловченко Р. И., Хачатрян В. Е. Алгоритм распознавания эквивалентности многоленточных автоматов без пересекающихся циклов 69
Полякова Т. И. Сложность оптимального по точности алгоритма вычисления сингулярного интеграла 72
Попов С. В. Об одном обобщении пропозиционального языка 74
Попов С. В., Трифонова Е. Е. Обучающая система по геометрии 76
Романов Д. С. Некоторые верхние оценки длин единичных проверяющих тестов относительно транспозиций переменных булевых функций 78
Рублев В. С., Дерябин В. О., Иоссель М. А., Карповский А. В., Лобачев Д. И., Юсупов А. Р. Классы отношений объектов и взаимодействия объектов 79
Садовник Е. В. О проверке на простоту чисел вида $N = 2kp^m - 1$ 80
Сапоженко А. А. Доказательство гипотезы Камерона-Эрдеша о числе множеств, свободных от сумм 82
Селезнева С. Н. О сложности поляризованных полиномов функций k-значных логик, зависящих от одной переменной 83
Сенченко А. С. Построение систем определяющих соотношений для автомата 84
Таранников Ю. В. О построении корреляционно-иммунных и устойчивых булевых функций 86

Тарасова В. П. О классах многоэкстремальных функций, допускающих поиск глобального экстремума методом Фибоначчи 87
Шалагин С. В. Дискретное моделирование преобразования квантового бита 89
Хелемендик Р. В. О выводе общезначимых формул из аксиом в логике ветвящегося времени 90
Чаплыгина Н. Б. Задача о равномерном назначении минимальной стоимости 92
Юсупов А. Р. Интерфейс пользователя динамической информационной модели DIM и навигатор объектов 94
Ярыкина М. С. Об оценках для l-упаковок большого радиуса 96
Яхонтов С. В. LINSPACE-конструктивность алгебраических чисел 98

Сложность квантовых моделей вычислений

Ф. М. Аблаев (Казань)

Модели вычислений, называемые квантовыми моделями вычислений, являются линейными моделями с дополнительными ограничениями, определяемыми постулатами квантовой механики. В математической кибернетике по аналогии с известными классическими классами сложности введены в рассмотрение квантовые классы сложности и установлены общие соотношения между ними и классическими классами.

- Нами доказано, что $PrQP = PP$. Здесь $PrQP$ (PP) класс сложности, содержащий языки, распознаваемые полиномиальными по времени квантовыми (классическими вероятностными) машинами Тьюринга с вероятностью $> 1/2$.

Мы предлагаем единый подход для определения классических и квантовой модели бинарных программ. С одной стороны бинарная программа – это контактная схема, с другой стороны в терминах бинарных программ естественным образом описываются машины Тьюринга и автоматы. Обозначим BP класс булевых функций, вычислимых детерминированными бинарными программами полиномиальной сложности. Известно, что $BP = LOGSPACE/poly$. Обозначим BP_w ($QuBP_w$) класс булевых функций, вычислимых классическими перестановчными детерминированными (квантовыми) бинарными программами полиномиальной сложности ширины w. Известно, что $BP_5 = NC^1$ и что $BP_2 \subset NC^1$.

- Нами доказано, что $QuBP_2 = NC^1$.
- Для один раз упорядоченно читающих бинарных программ (бинарных диаграмм решений, OBDD – в англоязычной литературе) нами показано, что булевская функция MOD_p представима в квантовых OBDD ширины $O(\log p)$, а классические (вероятностные и детерминированные) OBDD требуют ширины не менее p.

Работа выполнена при финансовой поддержке РФФИ, грант 03.01.00769

Сложность классического моделирования квантовых машин Тьюринга

Ф. М. Аблаев, А. Ф. Гайнутдинова (Казань)

В работе рассматриваются сложностные классы, определенные для модели машины Тьюринга (МТ). Обозначим $Time(T(n))$ – класс задач, решаемых на детерминированной МТ за время $T(n)$, $Space(S(n))$ – класс задач, решаемых на детерминированной МТ с использованием памяти $S(n)$. Аналогично определяются классы сложности $PrTime(T(n))$, $PrSpace(S(n))$ ($PrQTime(T(n))$, $PrQSpace(S(n))$) для вероятностных (квантовых) МТ, в которых вероятность правильного результата $p > 1/2$, и $BPTime(T(n))$, $BPSpace(S(n))$ ($BQTime(T(n))$, $BQSpace(S(n))$) – для вероятностных (квантовых) МТ, дающих правильный результат с большой вероятностью $p > 2/3$. При этом предполагается $T(n) \geq n$ и $S(n) \geq \log n$.

При моделировании квантовых вычислительных процессов используетс различная техника [2, 3]. В данной работе вычислительный процесс на классической вероятностной МТ

(PMT) и квантовой MT (QMT) рассматривается как линейный процесс. А именно, вычисление на PMT дл конкретного входа u является Марковским процессом, при котором на каждом шаге вектор распределения вероятностей конфигураций умножается на фиксированную стохастическую матрицу $M(u)$ для получения вектора распределения вероятностей не следующем шаге. Вычисление на QMT – это линейный унитарный процесс, аналогичный Марковскому процессу. Квантовый вычислительный шаг соответствует умножению чистого состояния (вектора распределения амплитуд конфигураций) на текущем шаге на фиксированную унитарную матрицу переходов $W(u)$ для получения вектора распределени амплитуд конфигураций на следующем шаге.

В данной работе мы показываем, что для унитарного линейного процесса можно построить эквивалентный (в смысле представления языков) Марковский процесс. Данная техника моделирования позволяет получить различные результаты по сложности моделировани квантовых вычислений. Как следствие из доказанной теоремы, мы имеем следующие соотношения сложностных классов:

Теорема. *1.* $PrQTime(T(n)) = PrTime(T(n))$, *в частности,* $PrQP = PP$.

2. $BQTime(T(n)) \subseteq PrTime(T(n))$ [1], *в частности* $BQP \subseteq PP$.

3. $BQSpace(S(n)) \subseteq PrSpace(S(n))$ [3] *и* $PrQSpace(S(n)) = PrSpace(S(n))$ [3].

Работа выполнена при поддержке гранта РФФИ 03-01-00769.

Литература

1. L. Adleman, J. Demarrais, M. Huang, Quantum computability, SIAM J. on Computing. 26(5), (1997), 1524–1540.

2. Bernstein and Vazirany, Quantum complexity theory, SIAM J. Comput, Vol. 26, No. 5, (1997), 1411–1473.

3. J. Watrous. Space-bounded quantum complexity. Journal of Computer and System Sciences , 59(2), (1999), 281–326.

Об оптимальном управлении динамическими структурами данных

Е. А. Аксенова, А. А. Лазутина
А. В. Соколов, А. В. Тарасюк (Петрозаводск)

Пусть мы работаем с двумя чистыми стеками [1,2,3] (разрешен доступ только к вершине стека), превосходящими объем быстрой памяти. Предполагается, что разрешено параллельное выполнение операций со стеками и заданы вероятности операций включения и исключения элементов одновременно в два стека. Ставится задача определить, в какое состояние следует переходить после обращения к памяти второго уровня, чтобы среднее время работы до перераспределения было максимально. В качестве модели рассмотрено блуждание по целочисленной решетке в треугольнике.

Пусть в памяти объема m единиц мы должны работать с тремя последовательными стеками. Двум стекам, растущим навстречу друг другу, отведем s единиц памяти, а третьему стеку $m-s$ единиц. В каждый момент времени с заданными вероятностями может произойти включение элемента одновременно в k стеков, где $0 \leq k \leq 3$ Нашей задачей является

выбор стека, который будет расположен отдельно, и нахождение такого значения s, чтобы среднее время до переполнения одного из стеков было максимальным. Можно показать [2], что эта задача сводится к оптимальному выбору детерминированных уровней заполнения урн в обратной урновой задаче в полиномиальной схеме размещения частиц [4], [5] по двум урнам, где одна урна - это пара стеков, растущих навстречу друг другу, а вторая - это отдельный стек. Здесь данная задача решается путем построения и численного решения разностного уравнения.

Рассматриваются задачи оптимального управления двумя последовательными циклическими FIFO очередями [1,6] в памяти одного уровня в случае последовательного и параллельного выполнения операций. Ставится задача определения оптимального размера памяти, выделяемого каждой очереди, для разных критериев оптимальности. Первая задача - максимизация среднего времени до переполнения одной из очередей. Вторая - минимизация доли времени нахождения в тех состояниях, в которых происходит потеря вставляемых в очередь элементов, при бесконечном времени работы. Этот критерий возникает, когда в случае переполнения одной из очередей работа не прекращается, а продолжается следующим образом: все элементы поступающие в переполненную очередь отбрасываются до тех пор, пока не появится свободная память. Такой способ работы с очередями применяется в сетевых маршрутизаторах и называется "сброс хвоста". В качестве модели мы имеем двумерное блуждание в прямоугольной области.

В докладе также обсуждаются вопросы построения модели нового метода управления очередями, предложенного в [7], когда очереди двигаются друг за другом по кругу, а также моделей, описывающих связанные методы представления динамических структур данных [1].

Работа выполнена при финансовой поддержке РФФИ, проект 01-01-00113.

Литература

1. Кнут Д. Искусство программирования для ЭВМ. Т. 1. М.: Мир, 1976.

2. Соколов А. В. Математические модели и алгоритмы оптимального управления динамическими структурами данных. Издательство Петрозаводского университета. 2002.

3. Аксенова Е. А., Волкова О. В., Лазутина А. А., Соколов А. В. Методы оптимального управления стеками в двухуровневой памяти // Труды Института прикладных математических исследований КарНЦ РАН. 2002. Вып. 3. С. 127–152.

4. Колчин В. Ф., Севастьянов Б. А., Чистяков В. П. Случайные размещения. М.: Наука, 1976.

5. Ивченко Г. И. Время ожидания и связанные с ним характеристики в полиномиальной схеме // Дискретная математика. 1993. Вып. 5. 3. С. 3–34.

6. Соколов А. В., Тарасюк А. В. Об оптимальном управлении циклическими очередями // Труды Института прикладных математических исследований КарНЦ РАН. 2002. Вып. 3. С. 190–195.

7. Соколов А. В. Об оптимальном управлении очередями в ограниченной памяти // Обозрение прикладной и промышленной математики. 2000. Т. 7. Вып. 2. С. 419–421.

Мощность максимального независимого множества для одного предиката на E_3

В. Б. Алексеев (Москва)

Пусть $E_3 = \{0, 1, 2\}$. Пусть $R(y, z)$ — любой двухместный предикат на E_3. Тогда предикат $R^n(\tilde{y}, \tilde{z})$ на E_3^n определяется следующим образом: если $\tilde{y} = (y_1, \ldots, y_n), \tilde{z} = (z_1, \ldots, z_n)$, то

$$R^n(\tilde{y}, \tilde{z}) \Longleftrightarrow \forall i \quad R(y_i, z_i).$$

Известно, что класс U_R функций f, сохраняющих предикат R, то есть удовлетворяющих условию

$$\forall \tilde{\alpha}, \tilde{\beta} \quad [R^n(\tilde{\alpha}, \tilde{\beta}) \Longrightarrow R(f(\tilde{\alpha}), f(\tilde{\beta}))],$$

является замкнутым. Всего в 3-значной логике P_3 имеется 74 попарно несимметричных типов двухместных предикатов [1]. Для большинства из них удается получить асимптотику (при $n \to \infty$) логарифма числа функций от n переменных, входящих в класс U_R. Однако для 6 предикатов эту задачу пока решить не удается. Одним из них является предикат $R_0(y, z)$, принимающий значение "ложь" только на парах (0,2),(2,0),(1,1). Для числа T_n функций от n переменных в классе U_{R_0} пока установлено только, что

$$\frac{1}{3}3^n \leq \log_2 T_n \leq \frac{1}{2}3^n(1 + o(1)).$$

Обычно при изучении классов типа U_R важную роль играет мощность максимального независимого множества, то есть такого множества M, что

$$\forall \tilde{\alpha}, \tilde{\beta} \quad [\tilde{\alpha} \in M, \tilde{\beta} \in M \Longrightarrow \neg R^n(\tilde{\alpha}, \tilde{\beta})].$$

Заметим, что предикат $R_0(y, z)$ симметричен. Поэтому предикат $R_0^n(\tilde{y}, \tilde{z})$ также симметричный. Мы доказываем следующее утверждение.

Теорема. *Пусть M — максимальное (по мощности) подмножество в E_3^n, независимое относительно предиката E_3^n. Тогда $|M| = \frac{1}{3}3^n$.*

Пример независимого множества в E_3^n дает множество $M = \{\tilde{\alpha} | \alpha_1 = 1\}$. Его мощность $|M| = \frac{1}{3}3^n$. Поэтому достаточно показать, что для любого независимого множества M в E_3^n выполняется $|M| \leq \frac{1}{3}3^n$. Доказательство этого вытекает из приводимых ниже лемм.

Для $\tilde{\alpha} \in E_2^n$ через $B_{\tilde{\alpha}}$ обозначим множество всех наборов из E_3^n, получающихся из $\tilde{\alpha}$ заменой любого подмножества нулей на 2. Будем говорить, что подмножество $N \subseteq E_2^n$ обладает свойством A_2, если

$$\forall \tilde{\alpha}, \tilde{\beta} \quad (\tilde{\alpha} \in N, \tilde{\beta} \in N \Longrightarrow \exists i(\alpha_i = \beta_i = 1)).$$

Лемма 1. *Существует максимальное независимое множество M, такое что*

$$M = \bigcup_{\tilde{\alpha} \in N} B_{\tilde{\alpha}}, \tag{1}$$

где $N \subseteq E_2^n$ и N удовлетворяет условию A_2.

Пусть $|\tilde{\alpha}|$ — вес набора $\tilde{\alpha}$ (число единиц в нем). Тогда из (1) получаем

$$|M| = \sum_{\tilde{\alpha} \in N} 2^{n - |\tilde{\alpha}|}. \tag{2}$$

Заметим, что любое множество M, задаваемое формулой (1), где N удовлетворяет A_2, является независимым в E_3^n относительно R_0^n. Поэтому мы приходим к следующей задаче на E_2^n : найти максимум функционала (2) по всем подмножествам $N \subseteq E_2^n$, удовлетворяющим свойству A_2.

Лемма 2. *Максимум функционала (2) по всем подмножествам $N \subseteq E_2^n$, удовлетворяющим свойству A_2, достигается на множествах $L_i = \{\tilde{\alpha} | \alpha_i = 1\}$ и только на них.*

В доказательстве леммы 2 используется то, что всякое подмножество в E_2^n, удовлетворяющее A_2, можно расширить до множества мощности 2^{n-1} с сохранением свойства A_2, что для любого подмножества N мощности 2^{n-1}, удовлетворяющего A_2, функция f_N такая, что $f_N(\tilde{\alpha}) = 1 \Leftrightarrow \tilde{\alpha} \in N$, является монотонной самодвойственной, а также то, что для подмножеств заданной мощности в слое (одного веса) куба E_2^n наименьшую тень имеет лексикографически начальный отрезок [2].

Из лемм 1 и 2 вытекает, что одним из максимальных независимых относительно R_0^n подмножеств в E_3^n является множество $M = \{\tilde{\alpha} | \alpha_1 = 1\}$. Но его мощность $|M| = \frac{1}{3}3^n$, откуда следует утверждение теоремы.

Данная работа поддержана грантом РФФИ 03-01-00783.

Литература

1. Баштанов Е. И. Типы двухместных отношений и их классов сохранения в трехзначной логике // Дискретная математика, т.1 , вып.2, 1989, с.86-96.

2. Kruskal F. B. The number of simplices in a complex // Mathematical Optimization Techniques, ed. by R. Bellman, 1963, p.251-278.

О сложности реализации функций k-значной логики поляризованными полиномами

В. Б. Алексеев, А. А. Вороненко, С. Н. Селезнева (Москва)

Рассматриваются полиномы относительно операций сложения и умножения по mod k. Известно [1], что если k – простое число, то любую функцию k-значной логики можно реализовать полиномом. В дальнейшем всюду предполагается, что k – простое число. Поскольку в этом случае $x^k \equiv x$, то можно считать, что показатели всех степеней в полиноме не превосходят $k-1$. При таком ограничении каждая функция k-значной логики реализуется единственным полиномом.

Пусть $P(x_1, \ldots, x_n)$ – полином, в котором t слагаемых с ненулевыми коэффициентами. Будем называть t сложностью полинома. Пусть $\pi_1, \ldots, \pi_n$ – произвольные подстановки на $E_k = \{0, 1, \ldots, k-1\}$. Выражение $P(\pi_1(x_1), \ldots, \pi_n(x_n))$, полученное из полинома P заменой вюду x_i на $\pi_i(x_i)$ $(i = \overline{1,n})$, будем называть поляризованным полиномом и сложностью его считать то же число t ненулевых слагаемых в P.

Пусть S – фиксированное множество подстановок на E_k. Поляризованной сложностью функции $f(x_1, \ldots, x_n)$ назовем минимальную сложность полинома, поляризованного какими-нибудь подстановками из S и реализующего функцию f. В [2] доказано, что для случая, когда S содержит только линейные сдвиги $\pi(x) = x + c (\mathrm{mod}\ k)$, выполняется

неравенство:

$$L_S(n) \leq (1 - \frac{1}{k(k-1)+1}) \cdot k^n.$$

Мы показываем, что имеет место следующее утверждение.

Теорема. *Для любого множества S подстановок на E_k выполняется асимптотическое (при $n \to \infty$) неравенство:*

$$L_S(n) \gtrsim (1 - \frac{1}{k}) \cdot k^n.$$

Доказательство использует мощностной метод. Если $|S| = R$, то число поляризованных полиномов сложности, не превосходящей r, не больше

$$R^n \cdot \sum_{t=0}^{r} C_k^t \cdot (k-1)^t.$$

Можно показать, что при $t = [\alpha \cdot k^n]$, где $\alpha = const$ и $\alpha < 1 - \frac{1}{k}$, при достаточно больших n выполняется неравенство

$$R^n \cdot \sum_{t=0}^{r} C_k^t \cdot (k-1)^t < k^{k^n}.$$

Откуда для любого $\alpha < 1 - \frac{1}{k}$ начиная с некоторого n выполняется неравенство $L_S(n) \geq \alpha \cdot k^n$, то есть

$$L_S(n) \gtrsim (1 - \frac{1}{k}) \cdot k^n.$$

Литература

1. Яблонский С. В. Введение в дискретную математику. М.: Высшая школа, 2001.

2. Селезнева С. Н. О сложности представления функций многозначных логик поляризованными полиномами. // Дискретная математика, т. 14, вып. 2, 2002, с. 48-53.

О надежности схем в базисе $\{\vee, \bar{}\}$ при неисправностях типа 0 на выходах элементов

М. А. Алехина (Пенза)

Речь идет о синтезе схем, реализующих булевы функции, с асимптотически наибольшей надежностью при однотипных константных неисправностях на выходах элементов. Получена нижняя оценка ненадежности схем в базисе $\{\vee, \bar{}\}$. Эта оценка достаточно точна – она асимптотически совпадает с верхней оценкой ненадежности схем, построенных автором в работе [1]. Показано, что почти каждая булева функция может быть реализована асимптотически наилучшей по надежности схемой, ненадежность которой асимптотически равна 2γ (γ – вероятность неисправности одного элемента) при $\gamma \to 0$.

Будем рассматривать реализацию булевых функций схемами из ненадежных элементов в базисе $\{\vee, \bar{}\}$ [2]. Схема реализует булеву функцию $f(x_1, \ldots, x_n)$, если при поступлении на входы схемы двоичного набора $\tilde{a} = (a_1, \ldots, a_n)$ при отсутствии неисправностей на выходе

схемы появляется значение $f(\tilde{a})$. Все элементы схемы независимо друг от друга переходят в неисправные состояния с вероятностью γ ($\gamma < 1/2$). Неисправности типа 0 на выходах элементов характеризуются тем, что в исправном состоянии функциональный элемент реализует приписанную ему булеву функцию, а в неисправном – константу 0. Аналогично определяются неисправности типа 1 на выходах функциональных элементов.

Пусть $P_{\bar{f}(\tilde{a})}(S, \tilde{a})$ – вероятность появления значения $\bar{f}(\tilde{a})$ на выходе схемы S, реализующей булеву функцию $f(\tilde{x})$ при входном наборе $\tilde{a}$. Ненадежность $P(S)$ схемы S определяется как максимальное из чисел $P_{\bar{f}(\tilde{a})}(S, \tilde{a})$ при всевозможных входных наборах $\tilde{a}$. Следовательно, надежность схемы равна $1 - P(S)$.

При однотипных константных неисправностях на выходах элементов произвольную булеву функцию нельзя реализовать схемой сколь угодно высокой надежности [3]. Возникает вопрос, какой максимальной надежности можно добиться при использовании ненадежных элементов, подверженных однотипным константным неисправностям на выходах? Ответ на него получен ниже.

Всюду далее предполагаем, что базисные элементы $x \vee y, \bar{x}$ подвержены неисправностям типа 0 на выходах элементов.

Теорема 1 [**1**]. При $\gamma \leq 1/130$ любую булеву функцию f можно реализовать такой схемой S, что $P(S) \leq 2\gamma + 18\gamma^2 + 415\gamma^3$.

Обозначим через $D(n)$ множество булевых функций вида $f(x_1, x_2, ..., x_n) = (x_{i_1}^{a_{i_1}} \vee ... \vee x_{i_k}^{a_{i_k}})^a$, где $i_j \in \{1, 2, ..., n\},\ j \in \{1, ..., k\}$.

Теорема 2 [**4**]. При $\gamma \leq 1/16$ функцию $f \in D(n)$ можно реализовать такой схемой S, что $P(S) \leq \gamma + \gamma^2 + 9\gamma^3$.

В работе [4] показано также, что любая схема S, содержащая хотя один функциональный элемент и реализующая функцию, отличную от константы, имеет ненадежность $P(S) \geq \gamma$ ($\gamma < 1/2$).

Следовательно, любую функцию $f \in D(n)$, кроме переменных и, быть может, констант, можно реализовать схемой, ненадежность которой асимптотически равна γ при $\gamma \to 0$. Эти схемы для названных функций являются асимптотически наилучшими по надежности.

Справедлива следующая теорема.

Теорема 3. *Пусть $f(\tilde{x})$– булева функция, $f \notin D(n)$, и S – любая схема, реализующая функцию f. Тогда при $\gamma < 1/2$ верно неравенство*

$P(S) \geq \gamma(1-\gamma)^m(1 + (1-\gamma)^{l-k} + \gamma(1-\gamma)^l)$, *где* $k \leq l,\ k, l, m \in \mathbf{Z}$.

Замечание. Нетрудно видеть, что

$\gamma(1-\gamma)^m(1 + (1-\gamma)^{l-k} + \gamma(1-\gamma)^l) \sim 2\gamma$ при $\gamma \to 0$.

Из теоремы 3 и замечания следует, что при $\gamma \leq 1/130$ любая схема, удовлетворяющая условиям теоремы 1 и реализующая булеву функцию $f(\tilde{x})$ ($f \notin D(n)$), является асимптотически наилучшей по надежности.

Найдем число фунций в классе $D(n)$

$$2(C_n^0 + C_n^1 + \sum_{k=2}^{n} C_n^k 2^k) = 2(1 + n + 3^n - 2n - 1) = 2 \cdot 3^n - 2n,$$

что мало по сравнению с общим числом 2^{2^n} булевых функций от n переменных.

Таким образом, при неисправностях типа 0 на выходах элементов $x \vee y$ и $\bar{x}$ почти все булевы функции в базисе из этих элементов можно реализовать схемами, ненадежность

которых асимптотически равна 2γ при $\gamma \to 0$. С точки зрения надежности функционирования эти схемы являются асимптотически наилучшими для почти всех функций.

Приведенные утверждения справедливы для двойственных функций в двойственном базисе $\{x\&y, \bar{x}\}$ при неисправностях типа 1 на выходах элементов [5]. Следовательно, при неисправностях типа 1 на выходах элементов $x\&y$ и $\bar{x}$ почти все булевы функции можно реализовать схемами, ненадежность которых асимптотически равна 2γ при $\gamma \to 0$. С точки зрения надежности функционирования эти схемы являются асимптотически наилучшими для почти всех функций.

В заключение автор благодарит проф. Н. П. Редькина за внимание к работе.

Литература

1. Алехина М. А. О надежности схем из ненадежных функциональных элементов при однотипных константных неисправностях на выходах элементов // Дискретная математика. 1993. Т. 5, вып. 2. С. 59–74.

2. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984.

3. Тарасов В. В. К синтезу надежных схем из ненадежных элементов // Матем. заметки. 1976. Т. 20, вып. 3. С. 391–400.

4. Алехина М. А. О надежной реализации функций специального вида в некоторых базисах при неисправностях типа 0 на выходах элементов // Материалы пятой молодежной научной школы по дискретной математике и ее приложениям (г. Москва, 12 – 18 ноября 2001 г.). М.: Изд-во центра прикл. исслед. при мех.-мат. ф-те МГУ, 2002. С. 9–14.

5. Алехина М. А. О надежности двойственных схем // Материалы XI Межгосударственной школы-семинара "Синтез и сложность управляющих систем" (г. Нижний Новгород, 20 – 25 ноября 2000 г.). Ч. 1. М.: Изд-во центра прикл. исслед. при мех.-мат. ф-те МГУ, 2001. С. 6–8.

О логарифмической выпуклости мощностей слоев k–значной n–мерной решетки

Т. В. Андреева (Москва)

В данной статье доказывается логарифмичесткая выпуклость мощностей слоев k–значной n–мерной решетки. Это свойство оказывается полезным при оценке мощностей слоев решетки (см. [1], [2]).

Множество $E_k^n = \{\tilde{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n) : \alpha_i \in \{0, 1, \ldots, k-1\}\}$ называется *k–значной n–мерной решеткой. Нормой набора* $\tilde{\alpha} \in E_k^n$ называется величина $\|\tilde{\alpha}\| = \sum_{j=1}^n \alpha_n$. Положим $F(i, n, k) = \{\tilde{\alpha} \in E_k^n : \|\tilde{\alpha}\| = i\}$. Множество $F(i, n, k)$ называется *i–ым слоем* E_k^n. Через $N(i, n, k)$ обозначим мощность множества $F(i, n, k)$. Результатом данной статьи является

Теорема *При любых $k \geq 2$, $n \geq 1$ и $2 \leq i \leq n(k-1)$ справедливо*

$$\frac{N(i-2, n, k)}{N(i-1, n, k)} \leq \frac{N(i-1, n, k)}{N(i, n, k)}. \tag{1}$$

Доказательство. Заметим, что для любых n и k при $0 \le i \le n(k-1)$ $N(i,n,k)$ равно количеству целочисленных решений уравнения

$$x_1 + x_2 + \cdots + x_n = i, \quad 0 \le x_i \le k-1. \tag{2}$$

Докажем выполнение (1) индукцией по n.

Очевидно, что если $n = 1$, то $N(i,1,k) = 1$ при любых k, $0 \le i \le k-1$, т.е. неравенство (1) выполнено.

Если $n = 2$, то $N(i,2,k) = i+1$ при $0 \le i \le k-1$, и $N(i,2,k) = 2k-i-1$ при $k \le i \le 2(k-1)$. Тогда имеем при $2 \le i \le k-1$

$$\frac{N(i-2,2,k)}{N(i-1,2,k)} = \frac{i-1}{i} < \frac{i}{i+1} = \frac{N(i-1,2,k)}{N(i,2,k)}.$$

При $i = k$ получаем $N(k-2,2,k) = N(k,2,k) = k-1$, тогда

$$\frac{N(k-2,2,k)}{N(k-1,2,k)} = \frac{k-1}{k} < \frac{k}{k-1} = \frac{N(k-1,2,k)}{N(k,2,k)}.$$

При $k+1 \le i \le 2(k-1)$ получаем

$$\frac{N(i-2,2,k)}{N(i-1,2,k)} = \frac{2k-i+1}{2k-i} < \frac{2k-i}{2k-i-1} = \frac{N(i-1,2,k)}{N(i,2,k)}.$$

Базис индукции доказан. Пусть (1) верно для всех $m \le n$, i и k. Докажем утверждение для $m = n+1$ и $k+1 \le i \le (n-1)(k-1)$, т.е. необходимо доказать неравенство

$$\frac{N(i-2,n+1,k)}{N(i-1,n+1,k)} \le \frac{N(i-1,n+1,k)}{N(i,n+1,k)}. \tag{3}$$

При $n \ge 3$ и $k-1 \le i \le (n-1)(k-1)$ имеем

$$N(i,n,k) = \sum_{j=0}^{k-1} N(i-j,n-1,k). \tag{4}$$

С использованием (4) получаем, что при $k+1 \le i \le (n-1)(k-1)$ неравенство (3) эквивалентно неравенству

$$1 - \frac{N(i-1,n,k) - N(i-2-(k-1),n,k)}{N(i-1,n+1,k)} \le 1 - \frac{N(i,n,k) - N(i-1-(k-1),n,k)}{N(i,n+1,k)}$$

или

$$N(i,n,k)\sum_{j=0}^{k-1} N(i-1-j,n,k) + N(i-1-k,n,k)\sum_{j=0}^{k-1} N(i-j,n,k) \le$$

$$\le N(i-1,n,k)\sum_{j=0}^{k-1} N(i-j,n,k) + N(i-k,n,k)\sum_{j=0}^{k-1} N(i-1-j,n,k).$$

Это справедливо, поскольку в силу индуктивного предположения при $0 \le j \le k-1$ и $k+1 \le i \le (n-1)(k-1)$ выполнены неравенства

$$\frac{N(i-1-j,n,k)}{N(i-j,n,k)} \le \frac{N(i-1,n,k)}{N(i,n,k)} \quad \text{и} \quad \frac{N(i-1-k,n,k)}{N(i-k,n,k)} \le \frac{N(i-1-j,n,k)}{N(i-j,n,k)}.$$

Неравенство (1) доказано при $k+1 \le i \le (n-1)(k-1)$.

Теперь докажем, что (1) выполняется при всех $n \ge 3$ и k для любых $2 \le i \le k$. При $0 \le i \le k-1$ из (2) получаем $N(i,n,k) = \binom{n+i-1}{i}$.

Следовательно,

$$\frac{N(i-2,n,k)}{N(i-1,n,k)} = \frac{i-1}{n+i-2} < \frac{i}{n+i-1} = \frac{N(i-1,n,k)}{N(i,n,k)}.$$

В случае, когда $i = k$, имеем из (2)

$$N(k,n,k) = \binom{n+k-1}{k} - n < \binom{n+k-1}{k},$$

отсюда следует, что

$$\frac{N(k-2,n,k)}{N(k-1,n,k)} = \frac{k-1}{n+k-2} < \frac{k}{n+k-1} < \frac{N(k-1,n,k)}{N(k,n,k)}.$$

Таким образом неравенство (1) доказано при $2 \le i \le k$.

Наконец, докажем, что (1) выполняется при всех $n \ge 3$ и k для любых $(k-1)(n-1)+1 \le i \le n(k-1)$. В этом случае уравнение (2) эквивалентно уравнению

$$y_1 + y_2 + \cdots + y_n = (k-1)n - i, \quad 0 \le y_i \le k-1, \quad \text{где} \quad y_i = k-1-x_i.$$

Тогда имеем $N(i,n,k) = \binom{(k-1)n-i+n-1}{n-1} = \binom{nk-i-1}{n-1}$, следовательно

$$\frac{N(i-2,n,k)}{N(i-1,n,k)} = \frac{nk-i+1}{nk-i+2-n} < \frac{nk-i}{nk-i+1-n} = \frac{N(i-1,n,k)}{N(i,n,k)}.$$

Тем самым теорема доказана.

Работа выполнена при поддержке гранта РФФИ 01-01-00266.

Литература

1. Т.В. Андреева, О мощности слоев трехзначной n–мерной решетки, Ж. вычисл. матем. и матем. физ., в печати.

2. Н.Н. Катериночкина, Некоторые соотношения для подмножеств слоев n–мерной k–значной решетки // Ж. вычисл. матем. и матем. физ., 1984, т. 24, N 5, с. 782–786.

Многочлены композиций и обобщенные многочлены Эйлера

Л. Н. Бондаренко (Пенза)

В работе методом производящих функций устанавливается связь между многочленами композиций, введенными в [1] соотношениями

$$C_n(D;z) = e_n^{-1} c_n(D;z), \text{ где } c_n(D;z) = \sum_{|\alpha|=n} e_\alpha z^\alpha, \tag{1}$$

и обобщенными многочленами Эйлера, используемыми в различных областях комбинаторного анализа. В (1) дифференциальный оператор Гельфонда–Леонтьева D определяется неубывающей последовательностью положительных чисел $\{d_k\}_1^\infty$ на полиномиальном базисе $\{t^k\}_0^\infty$: $D1{=}0$, $Dt^k{=}d_k t^{k-1}$, $k{\ge}1$. Этому оператору соответствует формальный степенной ряд $E(t,D){=}\sum_{k=0}^\infty e_k t^k$, где $e_0{=}1$, $e_k{=}(d_1 \cdots d_k)^{-1}$, являющийся обобщенной экспонентой. Набор комплексных чисел $z{=}(z_1,\ldots,z_n)$ в комбинаторных приложениях обычно составлен из нулей и единиц.

Под композицией $\alpha{=}(\alpha_1,\alpha_2,\ldots)$ понимается упорядоченное разбиение. Сумма всех частей композиции называется ее весом $|\alpha|{=}\alpha_1{+}\alpha_2{+}\ldots$, а число ненулевых членов — длиной

$\#\alpha$. Также используется обозначение $e_\alpha{=}e_{\alpha_1}\cdots e_{\alpha_n}$, $z^\alpha{=}z_1^{\alpha_1}\cdots z_n^{\alpha_n}$, а местоположение нулей в композиции α определяется с помощью зигзагообразного графа Мак–Магона [2].

Оператору D и последовательности комплексных чисел $\{z_n\}_1^\infty$ соответствует система обобщенных многочленов Гончарова $G_n(t,D;z)$

$$G_n(t,D;z){=}\sum_{k=0}^{n}(-1)^{n-k}g_{n,k}(D;z)\,e_k t^k\,, \tag{2}$$

биортогональная системе функционалов $\{D^n|_{t=z_{n+1}}\}_0^\infty$, причем

$$E(ts,D)=\sum_{k=0}^{\infty}E(z_k s,D)G_k(t,D;z)s^k\,. \tag{3}$$

В [1] получена связь между многочленами (1) и коэффициентами в (2)

Теорема 1. $2{\sum}'^{n}_{k=0}(-1)^k c_k(D;z)g_{n,k}(D;z)=\delta_{n0}$, где "$'$" у знака суммы означает умножение первого и последнего слагаемых на $1/2$, δ_{n0} — символ Кронекера, что при $n{\geq}0$ приводит к символическому равенству

$$2G_n(C,D;z)=c_n(D;z)+(-1)^n g_{n,0}(D;z)\,, \tag{4}$$

в котором при вычислении $G_n(C,D;z)$ полагается $C^k{\equiv}C_k(D;z)$.

Используя единичный набор z, из (3) формально выразим обыкновенную производящую функцию многочленов Гончарова через обобщенные экспоненты:

$$F_G(t,D,s){=}\sum_{k=0}^{\infty}G_k(t,D;z)s^k{=}E(ts,D)[E(s,D)]^{-1}\,,$$

а (4) заменим формулой

$$E(Cs,D)[E(s,D)]^{-1}{=}E(Cs,D)+[E(s,D)]^{-1}\,,$$

из которой найдем производящую функцию для чисел $C^k{\equiv}C_k(D)$:

$$E(Cs,D)=\frac{1}{2-E(s,D)}\,. \tag{5}$$

Используя параметр $u^{-1}{\in}(0,1]$, определим оператор $D(u^{-1})$ неубывающей последовательностью положительных чисел: $d_1,\{d_k u^{-1}\}_2^\infty$. Подставляя в соотношение (5) $D(u^{-1})$, получим $E(Cs,D(u^{-1})){=}E(\widetilde{C}(u)s,D)$,

$$\widetilde{C}_n(u)=e_n^{-1}\sum_{\#\alpha=1}^{n}e_\alpha u^{n-\#\alpha}\,, \tag{6}$$

$$\frac{1}{2-E(s,D(u^{-1}))}=\frac{u}{u+1-E(su,D)}\,,$$

где D, e_n, e_α соответствуют зачению параметра $u{=}1$.

Таким образом, замена в (5) оператора D на $D(u^{-1})$ приводит к экспоненциальной производящей функции для многочленов композиций (6), которая при замене u на $t-1$ становится производящей функцией для обобщенных многочленов Эйлера. Многочлены Эйлера

$A_n(t)$ степени $n-1$ [2, 3] используются, например, при изучении статистик симметрической группы перестановок S_n и определяются выражением:

$$A_n(t) = \sum_{\sigma \in S_n} t^{d(\sigma)} ,$$

где $d(\sigma)$ — число спусков перестановки $\sigma=\sigma_1\sigma_2 \ldots \sigma_n$, т. е. число всех $k<n$, для которых $\sigma_k>\sigma_{k+1}$.

Применение последовательности периода p (с одной единицей $z_1=1$ в периоде и остальными нулями) позволяет с помощью обобщенных гиперболических функций получить производящие функции, которые связаны, в частности, с updown-последовательностью, используемой в [4].

Литература

1. Бондаренко Л. Н. Оператор Гельфонда–Леонтьева и многочлены композиций // Материалы XII Международной школы–семинара "Синтез и сложность управляющих систем" (Пенза, 15–21 октября 2001 г.) Часть I. — М.: Изд–во центра прикладных исследований при механико-математическом факультете МГУ, 2001. — С. 47-53.

2. Риордан Дж. Введение в комбинаторный анализ. — М.: ИЛ, 1963.

3. Гульден Я., Джексон Д. Перечислительная комбинаторика. — М.: Наука, 1990.

4. Арнольд В. И. Исчисление змей и комбинаторика чисел Бернулли, Эйлера и Спрингера групп Коксетера // Успехи математических наук. —1992. — Т. 47, вып. 1. — С. 3-45.

Закономерности в деревьях вывода для стохастической разложимой КС-грамматики

А. Е. Борисов (Нижний Новгород)

В [2] для слов большой длины стохастического КС-языка установлены некоторые закономерности применения правил в выводе слов. При этом рассматривалась неразложимая грамматика, перронов корень которой строго меньше единицы (докритический случай). В настоящей работе аналогичные вопросы рассматриваются для одного класса разложимых грамматик.

Стохастическая контекстно-свободная грамматика [1] - это система $G = \langle V_T, V_N, R, s\rangle$, где V_T и V_N - множества терминальных и нетерминальных символов (дальше просто терминалы и нетерминалы), $s \in V_N$ - аксиома, R - конечное множество правил, $R = \bigcup\limits_{i=1}^{k} R_i$, где R_i - множество правил с одинаковой левой частью, имеющих вид

$$r_{ij} : A_i \stackrel{p_{ij}}{\to} \beta_{ij},\ j = 1, \ldots, n_i, \text{ где } A_i \in V_N, \beta_{ij} \in (V_T \cup V_N)^*,$$

а p_{ij} - вероятность применения правила r_{ij}.

Стохастическим КС-языком, соответствующим согласованной КС-грамматике G, называется множество слов, порожденных грамматикой с индуцированным на нем распределением вероятностей [1].

Первые моменты a_j^i грамматики определяются как математические ожидания числа нетерминалов A_j в правой части правил из R_i. Матрица $A = (a_j^i)$ называется матрицей первых моментов. Грамматика называется разложимой, если ее матрица первых моментов разложима.

Будем рассмотривать разложимую грамматику с двумя нетерминалами A_1, A_2, где A_1-аксиома, причем A_1 предшествует A_2 [3], и перронов корень матрицы меньше 1.

Матрица первых моментов такой грамматики имеет вид

$$A = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix},$$

причем $a, c < 1$. Дополнительно предположим, что $a, b, c > 0$. Пусть $u = (u_1, u_2)$, $v = (v_1, v_2)$ - правый и левый собственные вектора матрицы A для перронова корня r, при нормировке $u_1 v_1 + u_2 v_2 = 1$.

Через D^t обозначим множество деревьев вывода [1] высоты t. Рассмотрим случайную величину $q_{ij}(t, \tau)$ - число применений правила r_{ij} в деревьях из D^t на ярусе τ. Ее условное математическое ожидание обозначим $M_{ij}(t, \tau)$.

Теорема 1.

1) При $a \neq c$, $\tau \to \infty$, $t - \tau \to \infty$ выполняется асимптотическое равенство

$$M_{ij}(t, \tau) \sim p_{ij} \cdot \left(\frac{v_i}{r} \sum_{l=1}^{2} u_l s_l^{ij} + \sum_{l=1}^{2} u_l g_{il} \right), \ i, j \in \{1, 2\}$$

2) При $a = c = r$, $\tau \to \infty$, $t - \tau \to \infty$ для $j = 1, 2$ выполняются асимптотические равенства:

$$M_{1j}(t, \tau) \sim \frac{p_{1j} \cdot (t - \tau)}{t} \left(g_{11} + \frac{s_1^{1j}}{r} \right),$$

$$M_{2j}(t, \tau) \sim \frac{p_{2j}}{t} \left(\frac{\tau (s_2^{2j} + g_{22})}{r} + g_{21} \cdot (t - \tau) \right),$$

где g_{ij} - константы, определяемые вторыми моментами грамматики [3].

Заметим, что $M_{ij}(t, \tau)$ имеют для $a \neq c$ такой же вид, как и в неразложимом случае [2], а при $a = c$ величины $M_{ij}(t, \tau)$ линейно зависят от соотношения τ / t. При $t \to \infty$ правила, порождающие больше нетерминальных символов, используются чаще в выводе слов.

Далее, пусть $S_{ij}(t) = q_{ij}(t, 1) + q_{ij}(t, 2) + \ldots + q_{ij}(t, t)$ - число применений правила r_{ij} во всем дереве вывода. Рассмотрим величину $S_{ij}(t)/t$ - среднее число правил r_{ij} на одном ярусе дерева вывода.

Теорема 2.

1) Пусть $a \neq c$. Тогда

$$M(S_{ij}(t)/t) \to \omega_{ij} \text{ и } D(S_{ij}(t)/t) \to 0 \text{ при } t \to \infty, \ i, j \in \{1, 2\},$$

где $\omega_{ij} = p_{ij} \cdot \left(\frac{u_i}{r} \sum_{l=1}^{2} u_l s_l^{ij} + \sum_{l=1}^{2} u_l g_{il} \right)$.

2) Пусть $a = c$. Тогда

$$M(S_{1j}(t)/t) \to \omega_{1j}^1 / 2, \ M(S_{2j}(t)/t) \to (\omega_{2j}^1 + \omega_{2j}^2)/2,$$

и $D(S_{ij}(t)/t)$ *не стремится к* 0 *при* $t \to \infty,\ j = 1, 2$

(здесь $\omega^1_{1j} = p_{1j} \cdot (g_{11} + s_1^{1j}/r),\ \omega^1_{2j} = p_{2j} g_{12},\ \omega^2_{2j} = p_{2j} \cdot (g_{22} + s_2^{2j})/r$*).*

Таким образом, видно, что среднее число правил на один ярус дерева вывода не зависит от t при $t \to \infty$, как и в докритическом случае.

Литература

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1. М.: Мир, 1978.

2. Жильцова Л.П. Закономерности применения правил грамматики в выводах слов стохастического контекстно-свободного языка // Математические вопросы кибернетики. Вып.9. М.: Наука, 2000. С.101-126.

3. Севастьянов В.А. Ветвящиеся процессы. М.: Наука, 1971.

К вопросу о существовании стойких обфускаторов программ

Н. П. Варновский, В. А. Захаров (Москва)

Обфускатором (от англ. to obfuscate — сбивать с толку) называется специальный компилятор $\mathcal{O}$, который, получив на вход текст программы π, конструирует такую программу $\mathcal{O}(\pi)$, что

1). $\mathcal{O}(\pi)$ вычисляет ту же функцию, что и π;

2). размер и быстродействие $\mathcal{O}(\pi)$ не очень значительно отличаются от аналогичных показателей программы π;

3). для того, чтобы понять алгоритм, заложенный в программу $\mathcal{O}(\pi)$, требуется значительно больше усилий, чем в том случае, когда доступен текст исходной программы π.

Первые два требования легко формализовать, а вот формализация третьего требования (стойкости обфускации) — это непростая задача. В настоящее время имеется единственное математически строгое определение стойкости обфускатора [1]: обфускатор $\mathcal{O}$ считается стойким, если $\mathcal{O}(\pi)$ является «виртуальным черным ящиком», т.е. все, что можно узнать об алгоритме из текста программы $\mathcal{O}(\pi)$, можно эффективно вычислить на основе только тестовых экспериментов с программой $\mathcal{O}(\pi)$, не обращаясь к ее тексту. В [1] показано, что не существует универсальных обфускаторов, обладающих столь высокой стойкостью. Тем не менее, для многих приложений достаточно обфускаторов с более низкой стойкостью.

В настоящей заметке предпринята попытка дать альтернативное определение стойкости обфускации. Минимальное требование к стойкому обфускатору — это невозможность извлечения из текста $\mathcal{O}(\pi)$ значения одного-единственного предиката (функционального свойства программы). Такое требование можно записать в виде следующего определения.

Назовем *ансамблем* программ последовательность $\mathcal{H} = \{(S_n, D_n)\}_{n \in N}$, где S_n — выборка программ, а D_n — распределение вероятностей на S_n. *Секретным свойством* назовем предикат P, определенный на $S_{\mathcal{H}}$. Для каждого $n \in N$ предикат P является случайной величиной, заданной на S_n. Запись $\Pr_{\pi \xleftarrow{R} D_n}[P(\pi) = \sigma]$, $\sigma \in B$, обозначает вероятность того, что случайно выбранная программа π из S_n обладает ($\sigma = 1$) или не обладает ($\sigma = 0$) свойством P. Если для всякой пары программ π_1, π_2 их эквивалентность $\pi_1 \sim \pi_2$ влечет

$P(\pi_1) = P(\pi_2)$, то P называется *семантическим свойством*. Для простоты далее будем считать, что $\Pr_{\pi \xleftarrow{R} D_n}[P(\pi) = \sigma] = 1/2$.

Противником будем называть всякое множество $\mathcal{A}$ случайных полиномиальных алгоритмов. Каждый алгоритм из $\mathcal{A}$ получает на вход текст программы и вычисляет один бит. Если задан ансамбль $\mathcal{H}$, то результат применения противником $\mathcal{A}$ алгоритма для анализа обфускированных программ $\{\mathcal{O}(\pi) : \ \pi \in S_{\mathcal{H}}\}$ можно также рассматривать как случайную величину, определенную на вероятностном пространстве, включающем выборки программ. Запись $\Pr_{\pi \xleftarrow{R} D_n}[A(\mathcal{O}) = \delta]$, $\delta \in B$ будет обозначать вероятность вычисления алгоритмом A результата δ.

Компилятор $\mathcal{O}$ стойко обфускирует секретное свойство P относительно ансамбля $\mathcal{H}$ и противника $\mathcal{A}$, если для всякого алгоритма $A \in \mathcal{A}$ и для всякого натурального k выполняется $\lim\limits_{n\to\infty} \dfrac{\Pr\limits_{\pi \xleftarrow{R} D_n}(A(\mathcal{O}(\pi)) = P(\pi)) - 1/2}{n^k} = 0$.

Покажем, что в предположении существования односторонних перестановок данное требование стойкости удовлетворяется некоторыми обфускаторами в частных случаях. Рассмотрим ансамбль $\mathcal{H}_0 = \{(S_n, D_n)\}$, где $S_n = \{\pi_0\} \cup \{\pi_1^w : \ w \in \{0,1\}^n\}$, и распределение вероятности D_n определяется соотношениями $D_n(\pi_0) = 1/2$ и $D_n(\pi_1^w) = 1/2^{n+1}$. Программы ансамбля имеют следующий вид:

π_0: `input`(x); y:=0; `output`(y);

π_1^w: `input`(x); `if` x=w `then` y:=1 `else` y:=0; `output`(y);

Секретное свойство $P_0(\pi)$ задано предикатом $\exists x(F_\pi(x) = 1)$. Очевидно, $P_0(\pi_0) = 0$ и $P_0(\pi_1^w) = 1$.

Обфускация свойства $P_0(\pi)$ на ансамбле $\mathcal{H}_0$ проводится так. Предположим, что функция $\varphi(z)$ является односторонней перестановкой [2] и $h(z,u)$ является трудным предикатом [2]. Согласно [3] существование односторонней функции влечет существование трудных предикатов. Тогда компилятор $\mathcal{O}$ по заданной программе π из S_n вычисляет четверку (w,u,v,σ) и строит программу $\Pi^{w,u,v,\sigma}$ следующего вида:

`input(`x`); if` $\varphi(x)$=v `then if` $h(x,u)$=σ `then` y`:=0 else` y`:=1 else` y`:=0; output(`y`);`

Набор (w,u,v,σ) вычисляется так. Если $\pi = \pi_0$, то $\mathcal{O}$ выбирает случайную пару битовых строк w, u и полагает $v = \varphi(w)$, $\sigma = h(w,u)$. Если $\pi = \pi_1^w$, то $\mathcal{O}$ выбирает случайную битовую строку u и полагает $v = \varphi(w)$, $\sigma = 1 \oplus h(w,u)$.

Теорема 1. *Если односторонняя перестановка существует, то $\mathcal{O}$ стойко обфускирует секретное семантическое свойство P_0 относительно $\mathcal{H}_0$ и любого противника $\mathcal{A}_0$.*

Теорема 2. *Если секретное семантическое свойство P_0 допускает стойкую обфускацию относительно $\mathcal{H}_0$ и любого противника $\mathcal{A}_0$, то существуют односторонние функции.*

Приведенная здесь схема обфускации выделенного свойства заслуживает внимания в теоретическом и практическом аспектах: ее можно попытаться использовать как примитив для построения обфускаторов более высокой стойкости, а также для построения систем разграничения доступа (парольной защиты).

Настоящая работа выполнена при поддержке гранта РФФИ.

Литература

1. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K., On the (Im)possibility of obfuscating programs. *CRYPTO'01 — Advances in Cryptology*, Lecture

Notes in Computer Science, **2139**, 2001, p. 1-18.

2. Menezes A.J., Van Oorschot P.C., Vanstone S.A., *Handbook of Applied Cryptography.* CRC Press, 1997.

3. Goldreich O., Levin L.A., A hard-core predicate for all one-way functions. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, 1989, p. 25-32.

Об одной системе вывода, связанной со статическим анализом программ

М. С. Викторова, В. А. Захаров (Москва)

Основной задачей статического анализа програм является вычисление инвариантных соотношений между данными в выделенных точках программы с целью использования этих соотношений для последующей верификации программ. В работах [1,2] при решении этой задачи применялся метод абстрактных интерпретаций. Суть его состоит в том, что для заданной операционной семантики программ и анализируемого свойства строится абстрактная интерпретация программных операций и данных. В этой интерпретации данные образуют верхнюю полурешетку, и вычисление требуемых инвариантных соотношений сводится к вычислению итерационным методом верхних граней на множестве абстрактных данных, формирующихся каждым вычислением в выделенных точках программы.

Такой подход отличается универсальностью и во многих случаях приводит к удовлетворительным результатам. Однако он не свободен и от ряда недостатков, наиболее существенными из которых являются необходимость предварительного построения графа потока управления программы, невысокая эффективность итерационных алгоритмов, проявляющаяся при анализе некоторых свойств вычислений, а также трудности проведения вычисления с абстрактными интерпретациями на сложных моделях вычислений.

Одной из возможных альтернатив методу абстрактных интерпретаций служит подход к анализу программ, разработанны в [3] и получивший название логики потоков (Flow Logic). Основная идея этого подхода состоит в том, чтобы разделить задачу анализа на два этапа. Сначала осуществляется логический вывод системы ограничений, которым должны удовлетворять вычисляемые инвариантные соотношения. Для этой цели используется аппарат логического вывода. Далее для разрешения построенных систем ограничений используются специальные алгоритмы.

В настоящей работе предпринята попытка применить указанный подход к решению типичной задачи статического анализа — вычисления множеств инициализированных переменных. Переменная считается инициализированной в заданной точке программы, если при всяком вычислении, проходящем через эту точку, значение переменной полностью определяется самим вычислением.

Нами был рассмотрен упрощенный язык программирования, допускающий использование следующих операторов (x — переменная, t — терм, O — оператор):

1. `x :=` t (присваивание);

2. O_1`;` O_2 (последовательная композиция операторов);

3. `if` P `then` $\mathtt{O}_1$ `else` $\mathtt{O}_2$ `fi` (ветвление);

4. `while` P `do` `O` `od` (итерация).

Для большей наглядности результатов мы отказались от типизации переменных и сложных программных конструкций. Основное внимание было уделено построению системы вывода ограничений инициализированности переменных и обоснованию корректности и полноты построенной системы вывода. В качестве формул в предложенной системе вывода ограничений инициализированности переменных использовались выражения вида $I_0 \subseteq I_1$ и $I_0[O]I_1$, где I_0, I_1 — списки переменных, а O — один из указанных выше программных операторов. Содержательно, триаду $I_0[O]I_1$ следует истолковывать так: если все переменные из списка I_0 имеют определенные значения, то после выполнения оператора O все переменные из списка I_1 будут также иметь определенные значения.

Аксиомами исчисления служат все формулы вида $I_0 \subseteq I_1$, в которых I_0 является подсписком I_1. Правила вывода таковы.

$$\text{R1: } \frac{\vdash Var(t) \subseteq I_0}{\vdash I_0[\mathtt{x:=t}](I_0 \cup \{x\})}; \quad \text{R2: } \frac{\vdash \neg(Var(t) \subseteq I_0)}{\vdash I_0[\mathtt{x:=t}](I_0 - \{x\})};$$

$$\text{R3: } \frac{\vdash I_0[O_1]I_1 \ ; \vdash I_1[O_2]I_2}{\vdash I_0[O_1;O_2]I_2}; \quad \text{R4: } \frac{\vdash I_0[O_1]I_1 \ ; \vdash I_0[O_2]I_1}{\vdash I_0[\mathtt{if}\ P\ \mathtt{then}\ O_1\ \mathtt{else}\ O_2\ \mathtt{fi}]I_1};$$

$$\text{R5: } \frac{\vdash I_1 \subseteq I_0 \ ; \vdash I_1[O]I_1}{\vdash I_0[\mathtt{while}\ P\ \mathtt{do}\ O\ \mathtt{od}]I_1}; \quad \text{R6: } \frac{\vdash I_0' \subseteq I_0 \ ; \vdash I_0'[O]I_1' \ ; \vdash I_1 \subseteq I_1'}{\vdash I_0[O]I_1}.$$

Для приведенной системы удалось обосновать свойства корректности и полноты относительно абстрактных вычислений программ, а также построить разрешающий алгоритм, сводящий всякую формулу вида $I_0 \vdash O : I_1$ к системе теоретико множественных ограничений вида $I_0 \subseteq I_1$. Для изучения свойств корректности каждой программе O было сопоставлено множество $Tr(O)$ состоящее из конечного или бесконечного множества подстановок, отражающих результаты выполнения синтаксически допустимых вычислений программы. Переменная x считается инициализированной после выполнения оператора O, если для всякой подстановки $\theta \in Tr(O)$ терм $\theta(x)$ является основным. Множество переменных, инициализированных оператором O обозначим $Init(O)$.

Теорема. Для любого оператора O выполняется соотношение

$$I_1 \subseteq Init(O) \iff \vdash \emptyset[O]I_1 \ .$$

Работа выполнена при поддержке гранта РФФИ .

Литература

1. G. A. Kildall. A unified approach to global program optimization. In ACM Symposium on Principles of Programming Languages, 1973, p. 194–206.

2. P. Cousot, R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by constructions or approximation of fixpoints. In Proceedings of the ACM Symposium on the Principles of Programming Languages, 1977, p. 238–250.

3. F. Nielson, H. R. Nielson, C. Hankin. Principles of Program Analysis. Springer, ISBN 3-540-65410-0, 1999, 450 p.

Задача и алгоритм календарного планирования работ сервисного подразделения IT-компании

Т. В. Вознесенская, В. А. Костенко, М. И. Маркин (Москва)

Исходными данными для задачи календарного планирования работ сервисного подразделения IT-компании являются:

1. Типы планируемых работ. На множестве типов работ задано отношение частичного порядка, которое определяет допустимые последовательности выполнения работ.
2. Набор ресурсов (сотрудников подразделения). Каждый ресурс характеризуется: способностью выполнять те или иные типы работ, временем и стоимостью их выполнения, а также состояниями занятости (свободен, выполняет работу, болен, на тренинге, в отпуске) привязанными к временным интервалам.
3. Множество работ, которые необходимо запланировать. Для каждой работы заданы: тип, заказчик (характеризуется набором работ, выполненных для него ранее), удаленность (может увеличивать стоимость выполнени я работы и время выполнения работы), директивные сроки в пределах которых работа должна быть выполнена, штраф (за единицу времени) при невыполнении работы за директивные сроки, признак обязательности выполнения работы за директивные сроки.
4. Функции для вычисления прихода и расхода за выполнение работы.

Календарный план выполнения заданного множества работ определен, если для каждой работы из этого множества определены: ресурс, выделенный для выполнения работы, и сроки начала и завершения выполнения работы.

Критерием оптимальности календарного плана является разность между суммарным приходом и суммарным расходом по заданному множеству работ.

Календарный план корректен, если выполнены следующие условия:

1. один и тот же ресурс не может иметь перекрывающиеся по времени состояния занятости,
2. для любого заказчика не нарушены допустимые последовательности работ,
3. для работ, у которых есть признак обязательности выполнения за директивные сроки, не нарушены директивные сроки.

Основными особенностями задачи календарного планирования работ сервисного подразделения IT-компании являются: комбинаторный характер задачи и наличие разнородных ограничений.

Для решения данной задачи разработан жадный алгоритм планирования работ.

Суммарный приход за выполнения заданного набора работ определяется лишь типами работ и заказчиками и он для любого календарного плана одинаков. Поэтому, алгоритм строит календарный план так, чтобы минимизировать суммарный расход за выполнение заданного множества работ (расход зависит от назначения ресурсов на выполнение работ и нарушения директивных сроков выполнения работ). Работа алгоритма основана на следующих принципах:

1. Строится матрица расхода за выполнение работ с учетом исходно заданного состояния занятости ресурсов и без учета ограничения "один и тот же ресурс не может иметь перекрывающиеся по времени состояния занятости" при выполнении заданного набора работ. Элементы строки соответствуют расходам при выполнении одной работы различными ресурсами. Минимальный элемент строки определяет минимально возможный расход на выполнение работы.

2. Из минимальных элементов строк строится вектор минимального расхода за выполнение работ. Его размер равен числу заданных работ. Сумма элементов этого вектора является нижней гранью времени выполнения всех корректных календарных планов. В принципе она может быть не достижимой.

3. Последовательно просматриваются все элементы матрицы расхода за выполнение работ (условное назначение работы на ресурс) и для назначения выбираются работа и ресурс, которые минимально увеличивают сумму элементов вектора минимального расхода, из матрицы расходов исключается строка, соответствующая выбранной работе, корректируются столбец матрицы расходов, соответствующий выбранному ресурсу, и вектор минимального расхода. При этом учитывается (в отличии от п.1) ограничение "один и тот же ресурс не может иметь перекрывающиеся по времени состояния занятости" при выполнении заданного набора работ. Данные действия повторяются пока все работы не будут назначены на ресурсы.

Если данный алгоритм построил календарный план расход на выполнение которого равен сумме элементов первоначального вектора минимального расхода (построенног о в п.2), то календарный план является оптимальным. В противном верхняя оценка ошибки равна разнице между расходом на выполнение построенного календарного плана и суммой элементов первоначального вектора минимального расхода. Следует отметить, что при некоторых исходных данных верхняя оценка ошибки может быть сильно завышена.

Некоторые весовые и метрические свойства кодов Хэмминга

В. П. Воронин, Т. В. Трифонова (Москва)

В настоящей работе изучаются некоторые весовые и метрические свойства плотно упакованных кодов Хэмминга [1,2] как частично упорядоченных множеств с обычным для двоичных наборов отношением предшествования.

Пусть далее везде $n = 2^k - 1$ (k — натуральное). И пусть B^n — множество всех двоичных наборов $\tilde{\alpha}^n = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, у которых $\alpha_i \in \{0, 1\}$, $1 \leq i \leq n$, с обычным отношением предшествования, определенной для них метрикой Хэмминга и весом набора ($||\tilde{\alpha}^n||$). Все подробные определения и понятия можно найти в [3].

Множество всех наборов является n–мерным линейным пространством над полем $GF(2)$. Скалярное произведение наборов $\tilde{\alpha}^n$ и $\tilde{\beta}^n$ определяется так: $\left(\tilde{\alpha}^n, \tilde{\beta}^n\right) = \sum_{i=1}^{n} \alpha_i \cdot \beta_i$.

Пусть H — $(0, 1)$–матрица, содержащая k строк и n столбцов, каждый столбец которой является (при чтении сверху вниз) двоичным представлением номера столбца (с

незначащими нулями). Эту матрицу называют проверочной. Пусть далее код Хэмминга H^n — подпространство B^n, состоящее из всех наборов B^n, скалярное произведение каждого из которых на любую из строк проверочной матрицы H равно 0 (т. е. они ортогональны). Известно, что H^n является плотно упакованным линейным кодом с кодовым расстоянием 3.

Будем рассматривать H^n как частично упорядоченное множество с отношением предшествования, указанным выше. Пусть H_l^n — множество всех наборов из H^n, имеющих вес l, и $h(n,l)=|H_l^n|$. Заметим, что вместе с каждым набором в H^n входит ему противоположный, поэтому $h(n,l)=h(n,n-l)$.

Выведем рекуррентную формулу для вычисления $h(n,l)$. Очевидно, что $h(n,0)=1$, $h(n,1)=h(n,2)=0$. При выводе формулы будем $\tilde{\alpha}^n \in H_l^n$ интерпретировать как линейно зависимую совокупность столбцов проверечной матрицы, номера которых совпадают с номерами компонент $\tilde{\alpha}^n$, равных 1. Тогда справедливы следующие

Утверждения (приводимые вместе с обоснованиями) **:**

1) $h(n,3)=\dfrac{1}{3}\cdot\binom{n}{2}$.

Действительно, выберем произвольные два столбца и дополним их третьим столбцом, являющимся их покомпонентной суммой. Получим $\tilde{\alpha}^n \in H_3^n$. Эту совокупность можно было получить при выборе любых двух столбцов из получившихся трех, т. е. тремя способами.

2) $h\,(n,4)=\dfrac{1}{4}\cdot\left(\binom{n}{3}-h\,(n,3)\right)$.

Выберем произвольные три столбца. Возможны две ситуации.

i) Столбцы линейно зависимы, т. е. это $\tilde{\alpha}^n \in H_3^n$.

ii) Столбцы линейно независимы, при этом их сумма будет ненулевым столбцом, отличным от этих трех (иначе найдутся линейно зависимые два столбца из выбранных трех, а это невозможно). Так три столбца можно выбрать $\binom{n}{3}-h\,(n,3)$ способами. Дополним выбранные три столбца четвертым, являющимся их суммой. Получим $\tilde{\alpha}^n \in H_3^n$. Эту совокупность можно было получить привыборе любых трех столбцов из получившихся четырех, т. е. четырьмя способами.

3) При $2 \le l \le \dfrac{n-5}{2}$ справедливо равенство

$$h\,(n,l+2)=\frac{1}{l+2}\cdot\left(\binom{n}{l+1}-h\,(n,l+1)+(n-l)\cdot h\,(n,l)\right).$$

Выберем произвольные $l+1$ столбец. Возможны три ситуации.

i) Столбцы линейно зависимы, т. е. это $\tilde{\alpha}^n \in H_{l+1}^n$. Так выбрать $l+1$ столбец можно ровно $h(n,l+1)$ способами.

ii) Столбцы линейно независимы, при этом их сумма будет ненулевым столбцом, совпадающим с одним из выбранных. Удалив этот столбец, получим $\tilde{\alpha}^n \in H_l^n$. Так выбрать $l+1$ столбец можно ровно $(n-l)\cdot h\,(n,l)$ способами.

iii) Столбцы линейно независимы, при этом их сумма будет ненулевым столбцом, отличным от выбранных. Так выбрать $l+1$ столбец можно $\binom{n}{l+1}-h\,(n,l+1)-(n-l)\cdot h\,(n,l)$ способами. Дополним выбранные $l+1$ столбец $l+2$ столбцом, являющимся их суммой. Получим $\tilde{\alpha}^n \in H_{l+2}^n$. Эту совокупность можно было получить при выборе любых $l+1$

столбцов из получившихся $l+2$, т. е. $l+2$ способами, что и доказывает формулу.

Полученные рекуррентные формулы позволяют получить точное значение числа наборов из H_l^n, выраженное через определитель некоторой матрицы. Опишем эту матрицу.

Пусть $l \geq 3$, и Δ_l — квадратная матрица порядка $l-2$ следующего вида:

$$\delta_{1,i} = \frac{1}{(n-l+i)!} \text{ при } 1 \leq i \leq l-2, \quad \delta_{i,i} = 1 \text{ при } 2 \leq i \leq l-2,$$

$$\delta_{i+1,i} = 1 \text{ при } 1 \leq i \leq l-3, \quad \delta_{i+2,i} = (l-i)(n-l+i+1) \text{ при } 1 \leq i \leq l-4.$$

Все другие элементы матрицы, кроме выше определенных, равны 0. При $l \in \{3,4\}$ некоторые из описанных элементов в матрице отсутствуют.

Теорема 1. *Пусть* $l \geq 3$, *тогда* $h(n,l) = \frac{n!}{l!} \det \Delta_l$.

Доказательство. Для $l \in \{3,4\}$ в справедливости формулы можно убедиться, непосредственно вычислив определители первого и второго порядков соответственно.

При $l \geq 5$ разложим определитель по первому столбцу. Получим

$$h(n,l) = \frac{n!}{l!}\left(\frac{1}{(n-l+1)!}\det M_1 - \det \Delta_{l-1} + (l-1)(n-l+2)\det M_2\right).$$

Матрица M_1 — нижняя треугольная с единичной диагональю, поэтому $\det M_1 = 1$. Матрица M_2 во второй строке имеет единицу в первом столбце, а в остальных столбцах в этой строке стоят нули. Разложив по второй строке, получим $\det M_2 = \det \Delta_{l-2}$. В результате мы имеем полученную выше рукуррентную формулу. □

Непосредственное предшествование определяется так: $\tilde{\alpha}^n < \tilde{\beta}^n$, и нет таких $\tilde{\gamma}^n$, что $\tilde{\alpha}^n < \tilde{\gamma}^n < \tilde{\beta}^n$. Пусть A^n — множество всех атомов H^n, то есть наборов из H^n которым в H^n непосредственно предшествует нулевой набор.

Для любого $\tilde{\alpha}^n \in A^n$ справедливо $||\tilde{\alpha}|| \leq k+1$. Действительно, каждый атом можно рассматривать как линейно зависимую совокупность столбцов проверочной матрицы, в которой нет линейно зависимого собственного подмножества. Заметим, что $rank(H) = k$. Ясно также, что верхняя оценка достигается.

Пусть A_l^n — множество всех атомов A^n, имеющих вес l, и $a(n,l) = |A_l^n|$.Обозначим через $M(k,l)$ множество всех $(0,1)$–матриц ранга l, содержащих k строк и l столбцов, и $m(k,l) = |M(k,l)|$. Ясно, что при $l > k$ множество $M(k,l)$ пусто. Легко также убедиться, что

$$m(k,l) = (2^k-1)(2^k-2)(2^k-2^2)\ldots(2^k-2^{l-1}) = 2^{k\cdot l} \cdot \prod_{i=k-l+1}^{k}(1-2^{-i}).$$

Теорема 2. $a(n,l) = \frac{m(k,l-1)}{l!}$.

Доказательство. Рассмотрим произвольную матрицу из $M(k,l-1)$, добавим к $l-1$ столбцам этой матрицы l–ый столбец, являющийся суммой столбцов этой матрицы и лексикографически упорядочим их. Получим атом из H^n. Указанные действия обратимы. □

Оценим долю атомов наибольшего возможного веса, равного $k+1$, по отношению ко всем наборам из кода Хэмминга этого веса.

Теорема 3. $\dfrac{a(n,k+1)}{h(n,k+1)} \geq \dfrac{1}{4}$.

Доказательство. Так как $h(n,k+1) \leq \dfrac{1}{k+1}\binom{n}{k}$, то

$$\frac{a(n,k+1)}{h(n,k+1)} \geq \prod_{i=1}^{k} \frac{2^k - 2^{i-1}}{2^k - i} \geq 2^{\sum_{i=1}^{k-2} \log_2\left(1-2^{-i}\right)}.$$

Для $x \in \left[-\dfrac{1}{2}; 0\right]$ справедливо неравенство $\log_2(1+x) \geq 2x$. Заменив показатель степени на сумму членов геометрической прогрессии, получим нужное неравенство. □

В завершение заметим, что при $k \geq 4$ в A_3^n есть четыре атома, попарные расстояния между которыми равны 6, а в A_4^n — три атома, попарные расстояния между которыми равны 8, причем суммы этих совокупностей атомов равны одному и тому же набору $\tilde{\alpha}^n \in A_{12}^n$. Можно построить две неуплотняемые цепи из нулевого набора в $\tilde{\alpha}^n$ длины 4 и 5 соответственно, следовательно при $k \geq 4$ H^n не удовлетворяет цепному условию Жордана–Дедекинда. Аналогично устанавливается, что при $k \geq 4$ H^n не является решеткой.

Работа выполнена при поддержке гранта РФФИ 01-01-00266-а.

Литература

1. Hamming R. W. "Error Detecting and Error Correcting Codes", *Bell System Tech. J.*, **29**, 147–160 (1950); русский перевод: Хэмминг Р. В. "Коды с обнаружением и исправлением ошибок", в сб. "Коды с обнаружением и исправлением ошибок", ИЛ, М., 1956, с.7-22.

2. Петерсон У., Уэлдон Э. *Коды, исправляющие ошибки.* — М.: МИР, 1976. — 596с.

3. Гаврилов Г. П., Сапоженко А. А. *Задачи и упражнения по курсу дискретной математики.* — М.: Наука, 1992. — 408 с.

О нижних мощностных оценках функций Шеннона

С. В. Грибок (Москва)

Пусть задан некоторый класс S управляющих систем, реализующих булевы функции. Элементы класса S будем называть схемами. Будем предполагать, что для произвольной булевой функции $f(x_1, \ldots, x_n)$ найдется схема $\Sigma \in S$, реализующая эту функцию. Пусть каждой схеме $\Sigma \in S$ сопоставлено число $L_S(\Sigma) > 0$ – сложность схемы. Определим сложность $L_S(f)$ булевой функции f в классе S как минимальную из сложностей схем $\Sigma, \Sigma \in S$, реализующих f. Определим функцию Шеннона $L_S(n)$:

$$L_S(n) = \max_{f \in P_2^n} L(f),$$

где P_2^n –множество всех булевых функций от переменных $x_1, \ldots, x_n$. Обозначим через $g_S(L,n)$ число схем $\Sigma \in S$, реализующих булевы функции от переменных $x_1, \ldots, x_n$, и для которых $L(\Sigma) \leq L$.

Для вычисления нижних асимптотических оценок $L_S(n)$ в различных классах схем S обычно используется прием Риордана-Шеннона (см. [2]): вычисляется верхняя оценка числа схем

$$g_S(L,n) < \hat{g}_S(L,n), \tag{1}$$

и используя мощностное неравенство

$$g_S(L(n),n) \geq 2^{2^n},$$

выводится оценка для функции Шеннона

$$L_S(n) > \hat{L}_S(n). \tag{2}$$

В данной работе представлены явные формулы, выражающие коэффициенты формулы $\hat{L}_S(n)$ через коэффициенты формулы $\hat{g}_S(L,n)$. При этом $\hat{g}_S(L,n)$ и $\hat{L}_S(n)$ могут быть заданы со сколь угодно высокой степенью точности (см. например [1]). Использование этих формул позволяет выводить (2) из (1) непосредственно, избегая повторения однотипных вычислений.

Для многократных логарифмов, следуя [1], будем использовать следующие обозначения (основание 2 у логарифмов опускается):

$$l_{-1}(n) = 2^n, \quad l_0(n) = n, \quad l_{i+1}(n) = \log l_i(n), \quad i = 0,1,\ldots.$$

Теорема 1. *Если*

$$g_S(L,n) < \left(c \prod_{i=p}^{q} l_i^{\alpha_i}(L) l_{i-1}^{\beta_i}(n) \right)^L,$$

где p,q,c,α_i,β_i, $i = p,\ldots,q$, *зависят только от* S *и не зависят от* L *и* n, $0 \leq p < q$, $c > 0$, $\alpha_p + \beta_p > 0$, *то*

$$L_S(n) > \hat{L}_S(n),$$

где

$$\hat{L}_S(n) = \frac{\gamma_0 2^n}{l_p(n)} \left(1 + \sum_{i=p+1}^{q+1} \frac{\gamma_i l_i(n)}{l_p(n)} + \frac{\gamma_{q+2}}{l_p(n)} \right);$$

$$\gamma_0 = \frac{1}{\alpha_p + \beta_p}; \quad \gamma_{q+2} = \frac{\nu \log(\alpha_p + \beta_p) - \log c}{\alpha_p + \beta_p} - 1;$$

$$\gamma_i = \mu_i - \frac{\alpha_i + \beta_i}{\alpha_p + \beta_p}, \quad i = p+1,\ldots,q+1;$$

$\mu_i = 1$, *если* $i = 1$, *и* $\mu_i = 0$ *в противном случае;*
$\nu = 1$, *если* $p = 0$, *и* $\nu = 0$ *в противном случае.*
При этом доля тех функций $f(x_1,\ldots,x_n)$, *для которых*

$$L_S(f) \leq \hat{L}_S(n),$$

стремится к нулю с ростом n

Работа выполнена при финансовой поддержке РФФИ, грант 02-0101110.

Литература

1. Ложкин С.А. Оценки высокой степени точности для сложности управляющих систем из некоторых классов // Математические вопросы кибернетики, 1996, вып.6, с.189-214.

2. Лупанов О.Б. Асимптотические оценки сложности управляющих систем // Издательство Московского университета, 1984.

Сохранение топологических свойств среды при ее преобразованиях

В. И. Грунская (Ульяновск)

Проблема распознавания и отображения неизвестной среды является одной из фундаментальных проблем [1]. Ее естественной моделью является проблема исследования графа роботом. В настоящей работе в качестве среды рассмотрен плоский шахматный лабиринт [2], т.е. ориентированный симметричный граф, вершинами которого являются пары целых чисел, дуги параллельны осям x и y и естественным образом отмечены символами из множества {e, n, s, w} в соответствии с их направлением [3]. Вершины отмечены метками, указывающими их вид (внутренняя, угловая, граничная и т.д.)

Любой путь в лабиринте порождает последовательность отметок его вершин и дуг, названную протоколом. Множество протоколов путей, исходящих из вершины v обозначим L_v. Вершины v и u названы эквива-лентными, если L_v=L_u. В противном случае вершины v и u назовем отличимыми, а протокол из множества $L_v \backslash L_u$ различающим.

Введены преобразования подобия лабиринтов, аналогичные преобра-зованиям непрерывного евклидова пространства (параллельный перенос, осевая симметрия, поворот, гомотетия). В соответствии с ними введены преобразования подобия протоколов (ортогональность, инверсия, k-рас-тяжение).

Показано, что если лабиринт L' получен с помощью композиции преобра-зований подобия из лабиринта L, и протокол t различает вершины v и u лабиринта L, то протокол t', полученный с помощью композиции соответствующих преобразований из протокола t, различает их образы v' и u' в лабиринте L'.

Работа выполнена при поддержке РФФИ, грант 01-01-00080.

Литература

1. S. Albers and M.R. Henzinger. Exploring unknown environments. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 416-425, May 1997.

2. Кудрявцев В.Б., Ушчумлич Ш., Килибарда Г. О поведении автоматов в лабиринтах.//Дискретная математика.-1992.-Т.4,вып.3.-С.3-28.

3. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию конечных автоматов. - М.: Наука, 1985. - 320 с.

О контроле детерминированных графов

И. С. Грунский, С. В. Сапунов (Донецк)

Рассматривается задача контроля конечного связого неориентированного графа с помощью блуждающего по нему агента [1]. Агент перемещается по ребрам графа от вершины

к вершине и, находясь в вершине графа он считывает ее метку и метки смежных с ней вершин. Таким образом он может определить наличие или отсутствие некоторой метки в упомянутых вершинах. Задан класс всех таких графов с числом вершин не превосходящих число вершин графа-эталона и одним и тем же множеством меток. Требуется, имея полное описание графа-эталона найти множество путей по исследуемому графу, которое позволило бы определить изоморфен граф-эталон исследуемому графу или нет.

Рассмотрим множество всех конечных, простых, помеченных, связных, неориентированных графов. Пусть $G = (G, E, M, \mu, g_0)$ — такой граф, где G — множество вершин, E — множество ребер, M — множество меток, $\mu : G \to M$ — сюръективная функция размечивани вершин, g_0 — инициальная вершина, $E(g)$ — множество вершин, смежных с g.

Множество $O(g) = E(g) \cup \{g\}$ назовем окрестностью вершины g. Граф G назовем детерминированным если для любой вершины $g \in G$ и любых $s, t \in O(g)$ из $s \neq t$ следует $\mu(s) \neq \mu(t)$. В дальнейшем рассматриваются только детерминированные графы.

Последовательность меток вершин $\mu(g_1)...\mu(g_k)$, соответствующую некоторому пути $g_1...g_k$ в графе G назовем словом, порожденным вершиной g_1. Определим язык L_g как множество всех слов, порожденных вершиной $g \in G$. Две вершины $g, h \in G$ назовем неотличимыми, если $L_g = L_h$. L_{g_0} назовем языком порожденным графом G и обозначим его L_G. Два графа G и H назовем неотличимыми если $L_G = L_H$. Ясно, что если $L_G = L_H$, то для любой вершины $h \in H$ существует неоличимая от нее вершина $g \in G$. $G \cong H$ обозначает изоморфизм графов, сохраняющий начальную вершину и метки вершин.

Теорема 1 [2]. *Пусть $g, h \in G$. $L_g \neq L_h$ тогда и только тогда, когда существует слово $w \in L_g \oplus L_h$ и длина w не превосходит $|G| - |M| + 1$б причем эта оценка точна.*

Граф G назовем приведенным если $L_g \neq L_h$ для всех $g, h \in G$, $g \neq h$. Отношение неотличимости вершин является правой конгруэнцией и по ней определяется фактор-граф графа G.

Теорема 2 [2]. *Для любого детеминированного графа G существует единственный неотличимый от него граф с минимальным числом вершин изоморфный его фактор-графу.*

Пусть $\mathcal{K}$ – класс всех приведенных детеминированных графов с числом вершин не больше $|G|$, множеством меток M и $G \in \mathcal{K}$. Множество P конечных слов в алфавите M назовем контрольным экспериментом графа G относительно класса $\mathcal{K}$ (короче, для G и $\mathcal{K}$), если для любого графа $H \in \mathcal{K}$ и $H \ncong G$ найдется слово $p \in P$ и $p \in L_H \oplus L_G$. Другими словами P есть контрольный эксперимент для G и $\mathcal{K}$ если для любого $H \in \mathcal{K}$ из $L_H \cap P = L_G \cap P$ следует $G \cong H$. Пусть $[P]$ – пополнение множества P всеми начальными отрезками всех слов из P.

Теорема 3 [3]. *Для любого контрольного эксперимента P для G и $\mathcal{K}$, множество $[P] \cap L_G$ является обходом графа G по всем его ребрам.*

Покажем, что контрольный эксперимент всегда существует. Пусть $G = \{g_0, ..., g_{|G|-1}\}$. Множество слов $V = \{v_0, ..., v_{|G|-1}\}$ назовем базисом достижимости графа G если $g_0 \xrightarrow{v_i} g_i$. Множество слов $W = \{w_1, ..., w_k\}$ назовем базисом различимости графа G если для любых $g_i, g_j \in G$, $g_i \neq g_j$, существует $w \in W$ и $w \in L_{g_i} \oplus L_{g_j}$. Композиция $u \circ w$ слов u, w равна uw' если $u = u'm$, $w = mw'$, $m \in M$ и не определена в противном случае. Пусть e — пустое слово.

Теорема 4 [3]. *Множество слов $V(M \cup e) \circ W$ является контрольным экспериментом*

графа G, относительно класса $\mathcal{K}$.

Предложена процедура построения по контрольному эксперименту P специального дерева $T(P)$.

Теорема 5 [3]. *Если P является контрольным экспериментом для G и $\mathcal{K}$, то любой обход дерева $T(P)$ есть контрольный эксперимент для G и $\mathcal{K}$.*

Слово w назовем идентификатором [4] вершины $g \in G$ если для любого $h \neq g$, $h \in G$, $w \in W_g - L_h$.

Теорема 6. *1) Для любого $g \in G$ существует конечное множество $W_g \subseteq L_g$ и для любого $h \neq g$, $h \in G$, существует слово $w \in W_g - L_h$. 2) Слово $w^{(g)} = w_1 \circ (w_1)^{-1} \circ w_2 \circ (w_2)^{-1} \circ \ldots \circ w_n$ является идентификатором вершины $g \in G$, где $\{w_1, \ldots, w_n\} = W_g$ и $(w_i)^{-1}$ инверсия w_i.*

Предлагается следующий алгоритм построения контрольного эксперимента. Он базируетс на алгоритме Хенни [5] для автоматов Мили. Алгоритм состоит из двух частей. В первой части проводится контроль вершин графа в форме $w^{(0)} \circ T\left(g^{(0)}, g_1\right) \circ w^{(1)} \circ T\left(g^{(1)}, g_2\right) \circ \ldots \circ w^{(n-2)} \circ T\left(g^{(n-2)}, g_{n-1}\right) \circ w^{(n-1)}$, где $w^{(i)}$ – идентификатор g_i, $T\left(g^{(i)}, g_j\right)$ – слово соответствующее пути из $g^{(i)}$ в $g^{(j)}$, $g_i \overset{w^{(i)}}{\to} g^{(i)}$. Во второй части проводится контроль каждого ребра $(g_i, g_j) \in E$ посредством слова $T\left(g^{(j)}, g_i\right) \circ w^{(i)} \circ T\left(g^{(i)}, g_i\right) \circ w^{(i)}$.

Литература.

1. Levitt T., Lawton D.T. Qualitative navigation for mobile robot // Artifical Intelligence, 1990, - v.40. - p.305-360.

2. Сапунов С.В. Эквивалентность отмеченных графов // Труды ИПММ НАНУ, 2002, - т.7, - с. 162-167.

3. Сапунов С.В. Контроль детерминированных графов // Труды ИПММ НАНУ, 2002, - т.8, - с. 162-167.

4. Грунский И.С., Козловский В.А., Пономаренко Г.Г. Представления конечных автоматов фрагментами поведения – Киев: Наукова думка, 1990. – 230 с.

5. A. Bhattacharyya, Checking experiments in sequential machines, John Wiley & Sons, 1989.

Об одной системе линейных уравнений для стохастического КС-языка

Л. П. Жильцова (Нижний Новгород)

В задачах кодирования сообщений, являющихся словами некоторого языка, возникает необходимость в вычислении значений ряда величин, характеризующих эффективность кодирования. К таким величинам относятся, например, энтропия и стоимость кодирования.

В настоящей работе вопрос вычисления подобных величин рассматривается для стохастических контекстно-свободных языков (стохастических КС-языков) с однозначным выводом.

Стохастический КС-язык определяется с помощью стохастической КС-грамматики.

Стохастической КС-грамматикой называется система $G = \langle V_T, V_N, R, s\rangle$, где V_T и V_N — конечные множества терминальных и нетерминальных символов соответственно; $s \in V_N$ — аксиома, R — конечное множество правил, $R = \cup_{i=1}^{k} R_i$, где k — мощность алфавита V_N и $R_i = \{r_{i1}, \ldots, r_{i,n_i}\}$ — подмножество правил. Каждое правило в R_i имеет вид

$$r_{ij} : A_i \xrightarrow{p_{ij}} \beta_{ij}, j = 1, \ldots, n_i, \text{ где } A_i \in V_N, \beta_{ij} \in (V_T \cup V_N)^*$$

и p_{ij} — вероятность применения правила r_{ij}, удовлетворяющая условиям: $0 < p_{ij} \leq 1$ и $\sum_{j=1}^{n_i} p_{ij} = 1$.

Применение правила грамматики к слову в алфавите $(V_T \cup V_N)$ состоит в замене вхождения нетерминального символа из левой части правила на слово, стоящее в его правой части. КС-язык определяется как множество всех слов в алфавите V_T, каждое из которых может быть получено из аксиомы s с помощью конечного числа применений правил грамматики.

Каждому слову α КС-языка соответствует последовательность правил грамматики (вывод), с помощью которой α выводится из аксиомы s. Вероятность вывода определяется как произведение вероятностей правил, образующих вывод. Вероятность слова α определяется как сумма вероятностей всех различных левых выводов слова α (при левом выводе очередное правило применяется к самому левому нетерминалу в слове). КС-грамматика называется грамматикой *с однозначным выводом*, если каждое слово порождаемого языка имеет единственный левый вывод.

Грамматика G называется *согласованной*, если $\sum_{\alpha \in L} p(\alpha) = 1$. В работе рассматриваются согласованные КС-грамматики. Согласованная КС-грамматика G индуцирует распределение вероятностей P на множестве слов порождаемого КС-языка L и определяет *стохастический КС-язык* $\mathcal{L} = (L, P)$.

Определим матрицу A первых моментов грамматики следующим образом:

$$a_{im} = \sum_{j=1}^{n_i} p_{ij} s_m^{(ij)},$$

где $s_m^{(ij)}$ – число нетерминальных символов A_m в правой части правила r_{ij}. Элемент a_{im} – это математическое ожидание числа нетерминальных символов A_m в правых частях правил множества R_i.

Максимальный по модулю собственный корень матрицы A обозначим через r.

Сстохастическая КС-грамматика при отсутствии бесполезных нетерминалов (т.е. не участвующих в порождении слов языка) является согласованной тогда и только тогда, когда перронов корень матрицы первых моментов не превосходит единицы.

Поэтому, рассматривая КС-языки, будем предполагать, что $r \leq 1$.

Пусть L – язык, порожденный стохастической КС-грамматикой с однозначным выводом. На множестве правил R грамматики определим отображение

$$f : R \to \mathcal{R}^+,$$

где $\mathcal{R}^+$ – множество положительных вещественных чисел.

Пусть $\omega(\alpha) = r_{i_1 j_1} r_{i_2 j_2} \ldots r_{i_n j_n}$ – левый вывод слова $\alpha \in L$.

Определим $f(\alpha) = \sum_{l=1}^{n} f(r_{i_l j_l})$ и

$$M(f(L)) = \lim_{N\to\infty} \sum_{\alpha\in L, |\alpha|\le N} p(\alpha)\cdot f(\alpha)$$

(здесь $\mid \alpha \mid$ – длина слова α).

Пусть G – стохастическая грамматика с однозначным выводом, и A_i - некоторый нетерминальный символ. Через L_i обозначим язык, порожденный грамматикой G_i, которая получается из исходной грамматики G заменой аксиомы на нетерминал A_i. Таким образом, L_i есть множество слов в терминальном алфавите, выводимых из символа A_i.

Положим $L = L_1$ для исходного языка L. Через $M(f(R_i))$ будем обозначать $\sum_{j=1}^{n_i} p_{ij} \cdot f(r_{ij})$.

Теорема 1. *Пусть перронов корень r матрицы первых моментов порождающей грамматики G с однозначным выводом меньше 1. Тогда величины $M(f(L_i))$ конечны и удовлетворяют следующей системе линейных уравнений:*

$$M(f(L_i)) = M(f(R_i)) + \sum_{j=1}^{k} a_{ij}\cdot M(f(L_j)),\ (i = 1,\dots,k). \tag{1}$$

Теорема 2. *Пусть L – язык, порождаемый стохастической КС-грамматикой G с однозначным выводом, для которой перронов корень r матрицы первых моментов равен 1. Тогда $M(f(L))$ не ограничена.*

В качестве f рассмотрим отображение

$$f(r_{ij}) = -\log p_{ij}.$$

Очевидно, что $M(f(L_i))$ – энтропия языка L_i :

$$H(L_i) = \lim_{N\to\infty} \sum_{\alpha\in L_i, |\alpha|\le N} (-p(\alpha)\cdot \log p(\alpha))$$

и $M(f(R_i)) = H(R_i) = -\sum_{j=1}^{n_i} p_{ij}\cdot \log p_{ij}$ (логарифм берется по основанию 2).

Поэтому выполняется

Следствие 1. *При $r < 1$ величины $H(L_i)$ конечны и удовлетворяют следующей системе линейных уравнений:*

$$H(L_i) = H(R_i) + \sum_{j=1}^{k} a_{ij}\cdot H(L_j),\ (i = 1,\dots,k).$$

Рассмотрим отображение

$$F : r_{ij} \to v_{ij},\ \text{где } v_{ij} \in \{0,1\}^*. \tag{1}$$

Такое отображение используется в алгоритмах кодирования, когда для кодирования слова языка используется алфавитное кодирование последовательности правил в левом выводе [1].

По F определим отображение

$$f : r_{ij} \to \mid v_{ij} \mid,$$

где $\mid v_{ij} \mid$ – длина двоичной последовательности v_{ij}.

Величину $M(f(L_i))$ обозначим через $C(L_i(F))$ и назовем стоимостью кодирования F для языка L_i.

Следствие 2. *При $r < 1$ величины $C(L_i(F))$ конечны и удовлетворяют следующей системе линейных уравнений:*

$$C(L_i(F)) = C(R_i(F)) + \sum_{j=1}^{k} a_{ij} \cdot C(R_j(F)),\ (i = 1, \ldots, k).$$

Здесь $C(R_i(F)) = \sum_{j=1}^{n_i} p_{ij} \mid v_{ij} \mid$.

Наконец, рассмотрим в качестве f еще два отображения.

Отображение

$$f : r_{ij} \to 1$$

определяет математическое ожидание $M(f(L_i))$ длины левого вывода слова в языке L_i, которое обозначим через $\Omega(L_i)$, и $M(f(R_i)) = 1$.

Следствие 3. *При $r < 1$ величины $\Omega(L_i)$ конечны и удовлетворяют следующей системе линейных уравнений:*

$$\Omega(L_i) = 1 + \sum_{j=1}^{k} a_{ij} \cdot \Omega(L_j),\ (i = 1, \ldots, k)$$

Отображение

$$f : r_{ij} \to l_{ij},$$

где l_{ij} – число терминальных символов в правой части правила r_{ij}, определяет математическое ожидание $M(f(L_i))$ длины слова в языке L_i, которое обозначим через $D(L_i)$.

Следствие 4. *При $r < 1$ величины $D(L_i)$ конечны и удовлетворяют следующей системе линейных уравнений:*

$$D(L_i) = D(R_i) + \sum_{j=1}^{k} a_{ij} \cdot D(L_j),\ (i = 1, \ldots, k)$$

(здесь $D(R_i)$ – математическое ожидание числа терминальных символов в правой части правил множества R_i.)

Следствие 5. *При $r = 1$ величина $C(L(F))$ для любого кодирования вида (1), а также величины $H(L)$, $\Omega(L)$ и $D(L)$ не ограничены.*

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект N 01-01-00464).

Литература

1. Жильцова Л.П. О нижней оценке стоимости кодирования и асимптотически оптимальном кодировании стохастического контекстно-свободного языка // Дискретный анализ и исследование операций. Серия 1, т.8, N3. Новосибирск: Издательство Института математики СО РАН, 2001. С.26-45.

Об одной полисемантической модели последовательных программ

И. М. Захарьящев, В. А. Захаров (Москва)

Один из подходов к решению проблемы эквивалентности программ состоит в «утончении» отношения функциональной эквивалентности программ, что достигается построением аппроксимирующей модели программ, и в последующей разработке алгоритмов, разрешающих проблему эквивалентности схем программ для построеннной модели программ [1]. Семантика аппроксимирующей модели образуется за счет абстракции операционной семантики анализируемых программ; в простейшем случае в ее основе лежит система переходов (модель Крипке), состояния которой представляют абстрактные состояния данных, а переходы — имнтерпретацию исполняемых конструкций (операторов) программы. При этом синтаксическая структура самих программ не подвергается никаким изменениям. Вычисление каждой программ прокладывает две согласованные трассы: одну трассу в графе управления программы (схеме программы), а другую — в пространстве состояний данных. Состояния модели, через которые проходит вторая трасса истолковываются как промежуточные состояния данных вычисления. Две программы считаются эквивалентными на абстрактной модели, если, начав вычисление в одном и том же состояния, обе программы либо вообще не завершают вычисления, либо обе они завершают свои вычисления в одном и том же состоянии. Состояние системы переходов, достигнутое завершенным вычислением, истолковывается как результат вычисления.

Существуют, однако, такие разновидности программ и такие варианты проблемы эквивалентности программ, когда результат вычисления нельзя представить как последнее достигнутое промежуточное состояние данных. Примером могут служить императивные программы с операторами печати `output(x)` промежуточных результатов (здесь результат вычисления следует представлять в виде последовательности выделенных промежуточных состояний данных), а также рекурсивные программы де Беккера–Скотта (здесь результат вычисления образуется не только за счет «явных» преобразования данных, но также и и за счет символьных преобразований в самой программе — «ленивых» вычислений).

Для того, чтобы иметь возможность охватить в рамках единого подхода и описанные выше случаи проблемы эквивалентности программ, мы предлагаем обратиться к полисемантическим моделям программ, в которых допускается использование нескольких систем переходов (моделей Крипке) для интерпретации компонентов программы и проведения вычислений: в одной системе переходов задействованы промежуточные состояния данных, используемые для управления вычислением, а состояния другой системы истолковываются как окончательные результаты вычислений.

Пропозициональной схемой программы над алфавитами *базовых действий* $\mathcal{A}$ и *базовых условий* $\mathcal{C}$ называется система переходов

$$\pi = \langle \mathcal{P}_\pi, \mathbf{entry}, \mathbf{exit}, T_\pi, B_\pi \rangle,$$

где $\mathcal{P}_\pi$ обозначает множество процедур программы, **entry** и **exit** — вход и выход программы, $T_\pi\colon (\mathcal{P}_\pi \cup \{\mathbf{entry}\}) \times \mathcal{C} \to (\mathcal{P}_\pi \cup \{\mathbf{exit}\}$ — функция переходов, и $B_\pi\colon \mathcal{P}_\pi \cup \{\mathbf{entry}\} \times \mathcal{C} \to \mathcal{A}^*$ — функция операторов.

Интерпретация базовых действий и условий задается моделью $M = \langle \mathcal{F}, \mathcal{E}, \xi \rangle$, где $\mathcal{F} = \langle S, s_0, Q \rangle$ и $\mathcal{E} = \langle R, r_0, P \rangle$ — шкалы с множествами состояний S и R, начальными состоя-

ниями s_0 и r_0, и функциями преобразования данных $Q\colon S \times \mathcal{A} \to S$ и $P\colon R \times \mathcal{A} \to R$, а $\xi\colon S \to \mathcal{C}$ — функция оценки. Шкала $\mathcal{F}$ используется для интерпретации промежуточных состояний данных, а шкала $\mathcal{E}$ — для интерпретации результатов вычислений.

Вычислением схемы π на модели M называется такая конечная или бесконечная последовательность четверок

$$\rho = (\mathbf{entry}, s_0, c_0, t_0), (F_1, s_1, c_1, t_1), \ldots, (F_i, s_i, c_i, t_i), \ldots,$$

что для всякого $i \geq 0$ выполняется $c_i = \xi(s_i)$, $t_i = B_\pi(F_i, c_i)$, $F_{i+1} = T_\pi(F_i, c_i)$, $s_{i+1} = Q(s_i, t_i)$. Если $F_{n+1} = \mathbf{exit}$, то вычисление считается завершенным и состояние $R(r_0, t_0\ t_1 \ldots t_n)$ объявляется его результатом $r[\pi, M]$; если же вычисление бесконечно, то результат $r[\pi, M]$ не определен. Две схемы π_1 и π_2 называются эквивалентными на паре шкал $\mathcal{F}$, $\mathcal{E}$, если для всякой модели $M = \langle \mathcal{F}, \mathcal{E}, \xi \rangle$ выполняется $r[\pi_1, M] = r[\pi_2, M]$.

Теорема 1. *Если шкалы $\mathcal{F}$ и $\mathcal{E}$ порождаются свободной частично-коммутативной полугруппой базовых действий, то проблема эквивалентности программ на паре $(\mathcal{F}, \mathcal{E})$ разрешима за полиномиальное время.*

Возможность использования нового класса моделей программ для решения проблемы эквивалентности для традиционных моделей программ вытекает из следующего утверждения. Рассмотрим проблему эквивалентности линейных унарных рекурсивных программ на полугрупповых шкалах [2]. Тогда существует такая трансляция $\mathcal{T}$ линейных унарных рекурсивных программ в пропозициональные схемы программ, для которого справедлива

Теорема 2. *Для всякой шкалы $\mathcal{F}$, порожденной полугруппой базовых действий, существует такая шкала $\mathcal{E}$, также порожденная полугруппой базовых действий, что для любой пары линейных унарных рекурсивных программ π_1 и π_2 имеет место соотношение*

$$\pi_1 \sim_{\mathcal{F}} \pi_2 \iff \mathcal{T}(\pi_1) \sim_{\mathcal{F},\mathcal{E}} \mathcal{T}(\pi_2).$$

Настоящая работа выполнена при поддержке гранта РФФИ.

Литература

1. Подловченко Р.И. Моделирование программ схемами и построение систем преобразований схем. Кибернетика, 1982, N 6.

2. Захаров В.А. Об эффективной разрешимости проблемы эквивалентности линейных унарных рекурсивных программ. В сб. Математические вопросы кибернетики, вып. 8, 1999, с.255–273.

Приложение двойных неупорядоченных разбиений к решению одного нелинейного дифференциального уравнения второго порядка

И. И. Иванчик (Москва)

Назовем *однократным* неупорядоченным разбиением обычное неупорядоченное разбиение целого положительного числа p

$$m_1 + 2m_2 + 3m_3 + \ldots = p \tag{1}$$

и определим *двукратное* или, короче, *двойное* неупорядоченное разбиение пары чисел p, q

$$\sum_{s=1}\sum_{k=1+s} km_{k,s} = p, \qquad \sum_{s=1}\sum_{k=1+s} sm_{k,s} = q. \tag{2}$$

Разбиения (1) естественно возникают при разложении в ряд Тейлора по x функций $\phi(x)$ вида $\phi(x) = exp\left(\sum_k a_k x^k\right)$. Соответственно, разбиения (2) возникают при разложении в двойной ряд Тейлора по x, y функций $\phi(x, y)$ вида $\phi(x, y) = exp\left(\sum_s \sum_{k=1+s} a_{k,s} x^k y^s\right)$.

С помощью двойных неупорядоченных разбиений можно построить решение уравнения для плотности числа частиц $\nu(x)$ темной материи в астрофизике [1]

$$\frac{1}{x^2}\frac{d}{dx}\left(\frac{x^2}{\nu(x)}\frac{d\nu}{dx}\right) = \nu(x) - \sigma, \tag{3}$$

где $\sigma > 0$ – константа. Пусть $\nu_0 > 0$ – значение $\nu(x)$ в начале координат ($x = 0$). Анализ показывает, что $\nu(x)$ есть функция двух переменных, $\nu(x) = \nu(\zeta, \theta)$, где $\zeta = x\sqrt{\nu_0}$, $\theta = \frac{1}{6}\left(1 - \frac{\sigma}{\nu_0}\right)$, и решение надо искать путем разложения логарифма функции $\nu(\zeta, \theta)$ в двойной ряд Тейлора по ζ и θ:

$$\nu(\zeta, \theta) = \nu_0 exp\left(\theta\zeta^2 + \sum_{q=1}^{\infty}\sum_{p=1+q}^{\infty} \theta^q \zeta^{2p} C_{p,q}\right). \tag{4}$$

Уравнение (3) налагает на коэффициенты $C_{p,q}$ рекуррентные соотношения, зависящие от двойных неупорядоченных разбиений вида (2). Используя свойства двойных неупорядоченных разбиений, рекуррентные соотношения для коэффициентов ряда в показателе экспоненты (4) можно представить в виде

$q = 1$

$$C_{p,1} = \frac{3!}{(2p+1)!}, \tag{5}$$

$q = 2$

$$C_{p,2} = \frac{1}{2p(2p+1)}\left(C_{p-1,2} + h_{p-1,2} + \frac{3!}{(2(p-2)+1)!}\right), \tag{6}$$

$q = 3, 4, \ldots$

$$C_{p,q} = \frac{1}{2p(2p+1)}\left(C_{p-1,q} + \sum_{j=0}^{q-2}\frac{1}{j!}h_{p-1-j,q-j} + \sum_{j=1}^{q-2}\frac{1}{j!}C_{p-1-j,q-j} + \frac{3!}{(q-1)!\,(2(p-q)+1)!}\right), \tag{7}$$

где

$$h_{p,q} = \frac{1}{(2\pi i)^2}\oint\frac{d\psi}{\psi^{q+1}}\oint\frac{dz}{z^{2p+1}} exp\left(\sum_{s=1}^{q-1}\psi^s\sum_{k=1+s}^{\infty} z^{2k} C_{k,s}\right) \tag{8}$$

Рекуррентные соотношения в форме (5)–(8) позволяют преобразовать двойной ряд в показателе экспоненты (4) в однократный ряд по степеням θ

$$\nu(\zeta,\theta)=\nu_0 exp\left(\sum_{q=1}^{\infty}\theta^q\rho_q(\zeta)\right). \tag{9}$$

в котором коэффициентные функции $\rho_q(\zeta)$, $q=1,2,\ldots$ последовательно вычисляются при помощи (5)–(8) в аналитически замкнутом виде.

В качестве иллюстрации получаемых результатов приведем выражения для первых двух коэффициентных функций ($q=1,2$)

$$\rho_1(\zeta)=6\left(\frac{sh\zeta}{\zeta}-1\right), \tag{10}$$

$$\rho_2(\zeta)=\frac{(3!)^2}{2!}\int\limits_0^1 dt\, sh\zeta(1-t)\frac{\left(sh\zeta t-\zeta t-\frac{\zeta^3t^3}{3!}\right)^2}{\zeta t}+\left(\frac{3}{2}+\zeta^2\right)ch\zeta-\frac{195+2\zeta^2}{2}\frac{sh\zeta}{\zeta}+96+16\zeta^2+\frac{\zeta^4}{2}. \tag{11}$$

На описании структуры старших коэффициентных функций ($q=3,4,\ldots$) мы здесь не останавливаемся из-за отсутствия места.

Работа поддержана РФФИ, грант 01-01-00296

Литература

1. Иванчик И.И. К теории темной материи (в печати).

Базисы планарных графов

М. А. Иорданский (Нижний Новгород)

Рассматривается конструктивный подход к представлению графов, при котором одни графы, строятся из других с помощью операций, при выполнении которых производится отождествление изоморфных подграфов двух непересекающихся графов-операндов (операции бинарной *склейки*)[1]. Операции склейки относятся к одному *типу*, если отождествляемые при их выполнении подграфы изоморфны фиксированному графу. Операция склейки *сохраняет* некоторое *свойство* графов-операндов, если им обладает и результирующий граф.

Пусть P - множество графов, обладающих заданным свойством, H - система ограничений, обеспечивающая сохранение этого свойства при выполнении над графами операций склейки (H-склейки). Граф G называется H-*суперпозицией* графов из P, если $G\in P$ или граф G можно получить из графов множества P путем последовательного применения операций H-склейки; $[P]_H$-множество всех графов, являющихся H-суперпозициями графов из P. Если $[P]_H=P$, то P образует H-*замкнутый класс* графов. Минимальное по включению множество B_Θ графов из H-замкнутого класса P образует его *элементный базис*, если $[B_\Theta]_H=P$. Минимальное по включению множество B_O типов операций H-склейки, использования которых достаточно для построения из B_Θ всех графов H-замкнутого класса P образует его *операционный базис*. Поскольку типы операций склейки определяются

лишь видом отождествляемых подграфов, то операционный базисB_O задается их перечислением. Мощность элементного и операционного базиса определяет *сложность* конструктивного описания H-замкнутого класса графов.

Для класса $\mathcal{G}$ обыкновенных планарных графов рассматривается зависимость сложности конструктивного описания от ограничений, накладываемых на операции склейки, сохраняющие планарность. Используются обозначения: K_n - полный обыкновенный n-вершинный граф (K_0 - нуль-граф, не содержащий вершин); $\overline{G}$ - дополнение обыкновенного графа G до полного; C_n - простой цикл, содержащий $n \geq 4$ вершин; L_n - простая цепь с $n \geq 3$ вершинами.

Любая операция склейки сохраняет отсутствие петель, для сохранения отсутствия кратных ребер каждой паре несмежных отождествленных вершин результирующего графа должна соответствовать пара несмежных вершин хотя бы в одном из графов-операндов. Такие операции обозначаются как операции $\prec H \succ$-склейки. Отсутствие кратных ребер сохраняют, очевидно, также операции склейки по порожденным подграфам (операции $< H >$-склейки).

Каждый планарный граф допускает плоскую укладку, в которой вершины некоторой произвольной грани со связной границей расположены на окружности, вписанной в эту грань [1]. Если отождествляемые вершины графов-операндов принадлежат в их плоских укладках единым граням со связными границами и пары отождествляемых вершин выбираются в соответствии с круговыми обходами окружностей, вписанных в эти грани, то операции склейки ($H^{②}$-склейки) сохраняют планарность графов [1].

В работе [1] показано, что $\prec H^{②} \succ$-замкнутый класс $\mathcal{G}$ обыкновенных планарных графов имеет элементный базис $B_Э = \{K_1, K_2\}$ и операционный базис $B_O = \{K_0, \overline{K}_2\}$.

При переходе к операциям $\prec H_s^{②} \succ$-склейки, при выполнении которых в каждом из графов-операндов отождествляются не все вершины, имеем элементный базис $B_Э = \{K_1, K_2, K_3, K_4\}$ и операционный базис $B_O = \{K_0, K_1, \overline{K}_2, \overline{K}_3, \overline{K}_4, \overline{K}_5\}$ [2].

Для операций $< H_m^{②} >$-склейки, в которых отождествленные вершины образуют минимальное разделяющее множество в результирующем графе справедлива

Теорема. $< H_m^{②} >$*-замкнутый класс* $\mathcal{G}$ *обыкновенных планарных графов имеет элементный базис* $B_Э = \{K_1, K_2, K_3, K_4\}$ *и операционный базис* $B_O = \{K_0, K_1, \overline{K}_2, K_2, \overline{K}_3, (K_1 \circ K_2)K_0, L_3, K_3, \overline{K}_4, (\overline{K_2} \circ K_2)K_0,$
$(K_1 \circ L_3)K_0, (K_2 \circ K_2)K_0, L_4, C_4, L_5, C_5\}$.

При доказательстве полноты указанного множества типов операций склейки используются структурные свойства пятисвязных планарных графов [3]. Минимальность по включению обеспечивают графы правильных и полуправильных многогранников.

Сводная информация о мощности базисов класса обыкновенных планарных графов при всех вышеназванных ограничениях на операции склейки приведена ниже.

Ограничения / *Базисы*	$\prec H^{\textcircled{2}} \succ$	$\prec H_s^{\textcircled{2}} \succ$	$< H_m^{\textcircled{2}} >$	$< H_{m,e}^{\textcircled{2}} >$
$\lvert B_Э \rvert$	2	4	4	4
$\lvert B_O \rvert$	2	6	16	22

На основе подмножеств элементного и операционного базисов класса обыкновенных планарных графов можно задавать более узкие классы планарных графов.

Работа выполнена при финансовой поддержке РФФИ, грант 01-01-00464.

Литература

1. Иорданский М.А. Конструктивные описания графов//Дискретный анализ и исследование операций.1996.т.3,N4,С.35-63.

2. Иорданский М.А. Сложность конструктивных описаний планарных графов // Материалы IX Межгосударственной школы-семинара "Синтез и сложность управляющих систем"(Нижний Новгород,16-19 декабря 1998г.). - М.: Изд-во механико-математического факультета МГУ, 1999. - С.20-24.

3. Lebesgue H. Quelques consequences simple de la formule d'Euler // J. Math. Pures Appl. 1940. V.9. P.27-43.

Язык отчетов для динамической информационной модели DIM

М. А. Иоссель, В. С. Рублев (Ярославль)

Создание форм отчетов, как отмечено в [1], требует написания сложных SQL-запросов. Для того, чтобы пользователь сам мог в большинстве случаев построить сложные формы отчетов с включением статистики и других вычислений необходима разработка языка отчетов и формул, а также на этой базе генератора отчетов.

Отчет может иметь сложную форму, которая диктуется в свою очередь сложной иерархической структурой данных. В этом случае требуется переход в отчете от одного уровня подробности данных к другому. Другая проблема состоит в выборе данных объекта, одни части которого находятся в различных отношениях с другими частями. Эффективность выбора зависит от конструкций языка, позволяющих быстро находить все связанные данные. Третья проблема состоит в накоплении итогов, так как они должны

готовиться одновременно для разных уровней подробности данных и выдаваться в отчет в соответствующих местах. И, наконец, четвертая проблема состоит в необходимости делать сложные вычисления при накоплении итогов и их окончательной обработке.

Все указанные проблемы решаются определенным способом организации формы отчета, языка определения данных и формул. Организация отчета состоит в делении его на ряд разделов, соответствующих уровням подробности данных. Иерархические связи данных приводят к структуре отчета в виде дерева. Каждый раздел отчета может состоять из трех частей: *шапка, тело* и *концовка* раздела (обязательным является только *тело* раздела). В *теле* раздела содержатся имена параметров объектов, которые возможно подлежат выводу в отчет. При этом каждый параметр определен своим *именем* и *именем класса*, а также возможно связан с другими данными отчета, что отмечается языковыми конструкциями. Предполагается перебор данных по таблицам связанных объектов с ограничителями, указанными в параметрах *шапки* раздела.

Параметры *шапки* каждого раздела либо определяются в иерархически старших по дереву разделах, либо задаются внешними параметрами отчета. Во время перебора данных раздела происходит накопление итогов этого раздела и, возможно, всех разделов, содержащих данный. Потребность в выводе итогов определяется *концовкой* раздела. В качестве итогов могут выступать как простейшие функции от данных раздела (*количество значений, сумма, минимум, максимум*), так и сложные формулы от других итогов, не выводимых в *концовке*. Отметим, что данные тела раздела также могут не выводится, а служить лишь для накопления итогов (и, возможно, не для самого раздела, а для содержащих его разделов).

Реализация формы отчета делается при помощи таблицы формы отчета, в которой для каждого раздела и части описывается текст, указатели данных, функции и формулы.

При генерации отчета алгоритм обрабатывает форму, обходя дерево разделов от корня к листьям и слева направо. При этом последовательно сгенерированные *шапка, тело* и *концовка* раздела выдаются в специальную таблицу отчета.

Вывод отчета из этой таблицы может быть сделан многочисленными средствами форматирования.

Дальнейшее направление исследований связано с введением в генератор отчета интерактивных средств, позволяющих простому пользователю создавать отчеты при помощи визуальных средств. При этом выбор тех или иных средств должен отображаться автоматически при помощи языка отчетов в соответствующие разделы формы.

Литература

1. Рублев В. С., Дерябин В. О., Лобачев Д. И., Юсупов А. Р. Базовые отношения обектов баз данных и гибкие таблицы //Моделирование и анализ информационных систем. Ярославль, 2002. Т.9, N 2. С.16-27.

Каркасная классификация помеченных блоков

Г. И. Калмыков (Москва)

Цель работы — описать каркасную классификацию неразделимых помеченных графов

(помеченных блоков) со степенью вершин не менее числа 3. Мы будем придерживаться терминов, введенных в [2].

О п р е д е л е н и е 1. Блок, в котором степень всех вершин не менее числа 3, называется *3-блоком.*

Рассмотрим ансамбль каркасных циклов [2] $\mathbf{C}$, содержащий хотя бы один 3-блок. Мы будем предполагать, что все вершины этих циклов принадлежат множеству $V_n = \{1, 2, \ldots, n\}$ и каждая вершина из множества V_n принадлежит по крайней мере одному из этих циклов. Через $V_0(\mathbf{C})$ обозначим множество всех вершин блока $S(\mathbf{C})$ [2], имеющих степень вершин не менее числа 3. Пусть $v \in V_n$. Через $(i(v))_{q(v)}$ обозначим вектор, удовлетворяющий следующим двум условиям: (1) вершина v принадлежит циклу $C(i(v))_{q(v)}$; (2) вершина v не принадлежит ни одному из циклов ансамбля $\mathbf{C}$, помеченных векторами, предшествующими вектору $(i(v))_{q(v)}$. Через $U_1(\mathbf{C}; v)$ мы будем обозначать множество всех вершин, предшествующих вершине v по циклу $C(i(v))_{q(v)}$ и не входящих в множество $V_0(\mathbf{C})$.

Обозначим через $\Omega(\mathbf{C}) = \{\overline{\omega} = (\omega^{(1)}, \omega^{(2)}, \omega^{(3)})\}$ совокупность, в которой каждый элемент $\overline{\omega} = (\omega^{(1)}, \omega^{(2)}, \omega^{(3)})$ представляет собой тройку отображений $\omega^{(1)}$, $\omega^{(2)}$, $\omega^{(3)}$. Здесь $\omega^{(1)}$ есть отображение множества вершин $V_1(\overline{\omega})$, входящего в множество вершин $V_n \setminus V_0(\mathbf{C})$, в совокупность множеств $\mathcal{U}''(\overline{\omega}) = \{U\}$, состоящую из всех подмножеств (включая и пустое множество) множества вершин $V''(\overline{\omega}) \subseteq V_1(\overline{\omega})$. $\omega^{(2)}$ есть отображение множества вершин $V_2(\overline{\omega})$, входящего в множество вершин $V_n \setminus (V_0(\mathbf{C}) \bigcup V_1(\overline{\omega}))$, в совокупность множеств $\mathcal{U}' = \{U\}$, состоящую из всех непустых подмножеств множества вершин $V'(\overline{\omega}) = V_1(\overline{\omega}) \setminus V''(\overline{\omega})$. Отображение $\omega^{(3)}$ определено на множестве $V_3(\overline{\omega})$. Если $V_n \neq V_0(\mathbf{C}) \bigcup V_1(\overline{\omega}) \bigcup V_2(\overline{\omega})$, то $V_3(\overline{\omega}) = V_n \setminus (V_0(\mathbf{C}) \bigcup V_1(\overline{\omega}) \bigcup V_2(\overline{\omega}))$, а $\omega^{(3)}$ есть отображение множества вершин $V_3(\overline{\omega})$ в множество $V_0(\mathbf{C}) \bigcup V_1(\overline{\omega})$; в противном случае $V_3(\overline{\omega}) = \{\emptyset\}$, а $\omega^{(3)}(\emptyset) = \emptyset$.

Пусть $\overline{\omega}$ — тройка отображений из множества $\Omega(\mathbf{C})$. Введем обозначения:

$$U_3(\overline{\omega}; v) = U_1(\mathbf{C}; v) \bigcap V_2(\overline{\omega}); \quad U_5(\overline{\omega}; v) = U_1(\mathbf{C}; v) \bigcap V_1(\overline{\omega});$$

$$U_4(\overline{\omega}; v) = \bigcup_{u \in U_3(\overline{\omega}; v)} \omega^{(2)}(u); \quad U_6(\overline{\omega}; v) = \bigcup_{u \in U_5(\overline{\omega}; v)} \omega^{(1)}(u).$$

Обозначим через $\Omega_0(\mathbf{C}) = \{\overline{\omega} = (\omega^{(1)}, \omega^{(2)}, \omega^{(3)})\}$ множество, состоящее из всех тех элементов совокупности $\Omega(\mathbf{C}) = \{\overline{\omega} = (\omega^{(1)}, \omega^{(2)}, \omega^{(3)})\}$, которые удовлетворяют следующим десяти условиям: (A_1) $\omega^{(1)}(V_1(\overline{\omega})) = V''(\overline{\omega})$. (A_2) $\omega^{(2)}(V_2(\overline{\omega})) = V'(\overline{\omega})$. (A_3) Если v' и v'' — две различные вершины из множества $V_1(\overline{\omega})$, то их образы $\omega^{(1)}(v')$ и $\omega^{(1)}(v'')$ не имеют общих вершин. (A_4) Если v' и v'' — две различные вершины из множества $V_2(\overline{\omega})$, то их образы $\omega^{(2)}(v')$ и $\omega^{(2)}(v'')$ не имеют общих вершин. (A_5) Образ $\omega^{(2)}(v)$ любой вершины $v \in V_2(\overline{\omega})$ не содержит ни одной вершины, принадлежащей множествам $\omega^{(2)}(U_3(\overline{\omega}; v))$ и $U_1(\mathbf{C}; v)$. (A_6) Образ $\omega^{(1)}(v)$ любой вершины $v \in V_1(\overline{\omega})$ не содержит ни одной вершины, принадлежащей множествам $V'(\overline{\omega})$, $U_1(\mathbf{C}; v)$ и $\omega^{(1)}(U_5(\overline{\omega}; v))$. (A_7) Каждая вершина, принадлежащая множеству $\omega^{(1)}(v)$, где $v \in V_1(\overline{\omega})$, соединяется с вершиной v ребром, принадлежащим множеству $X_{ad}(\mathbf{C})$ (см. [2]). (A_8) Каждая вершина, принадлежащая множеству $\omega^{(2)}(v)$, где $v \in V_2(\overline{\omega})$, соединяется с вершиной v ребром, принадлежащим множеству $X_{ad}(\mathbf{C})$. (A_9) Функция $\omega^{(3)}$ удовлетворяет условию: какова бы ни была принадлежащая множеству $V_3(\overline{\omega})$ вершина v, ее образ $\omega^{(3)}(v)$ принадлежит либо множеству $V_0(\mathbf{C})$, либо множеству $U_4(\overline{\omega}; v) \setminus U_1(\mathbf{C}; v)$ а ребро $\{v, \omega^{(1)}(v)\}$ принадлежит

множеству $X_{ad}(\mathbf{C})$. (A_{10}) Если вершина v принадлежит множеству $V''(\overline{\omega})$, то эта вершина принадлежит множеству $U_6(\overline{\omega}; v)$.

Определение 2. Элементы совокупности $\Omega_0(\mathbf{C}) = \{\overline{\omega} = (\omega^{(1)}, \omega^{(2)}, \omega^{(3)})\}$ называются *триплетами отображений* ансамбля циклов $\mathbf{C}$.

Пусть $\overline{\omega}$ — триплет из совокупности $\Omega_0(\mathbf{C})$. Введем обозначения:

$$X_1(\overline{\omega}; v) = \{\{u, v\} : u \in \omega^{(1)}(v)\},\ v \in V_1(\overline{\omega}); \quad X_2(\overline{\omega}; v) = \{\{u, v\} : u \in \omega^{(2)}(v)\},\ v \in V_2(\overline{\omega});$$

$$X_1(\overline{\omega}) = \bigcup_{v \in V_1(\overline{\omega})} X_1(\overline{\omega}; v); \quad X_2(\overline{\omega}) = \bigcup_{v \in V_2(\overline{\omega})} X_2(\overline{\omega}; v); \quad X_3(\overline{\omega}) = \bigcup_{u \in V_3(\overline{\omega})} \{u, \omega^{(3)}(u)\};$$

$$X(\overline{\omega}) = X_1(\overline{\omega}) \bigcup X_2(\overline{\omega}) \bigcup X_3(\overline{\omega});$$

$\tilde{G}(\overline{\omega})$ — граф с множеством вершин V_n и множеством ребер $X(\overline{\omega})$; $S(\mathbf{C}; \overline{\omega}) = S(\mathbf{C}) \bigcup \tilde{G}(\overline{\omega})$.

Для каждой вершины $v \in V_1(\overline{\omega})$ определим множество вершин $W_1(\overline{\omega}; v)$ и множество ребер $Y_1(\overline{\omega}; v)$, полагая

$$W_1(\overline{\omega}; v) = (U_6(\overline{\omega}; v) \bigcup V'(\overline{\omega})) \setminus U_1(\mathbf{C}; v); \quad Y_1(\overline{\omega}; v) = \{\{u, v\} : u \in W_1(\overline{\omega}; v)\}.$$

Для каждой вершины $v \in V_2(\overline{\omega})$ определим множество ребер $Y_2(\overline{\omega}; v)$, полагая $Y_2(\overline{\omega}; v) = \{\{u, v\} : u \in U_4(\overline{\omega}; v) \setminus U_1(\mathbf{C}; v)\}$. Для каждой вершины $v \in V_3(\overline{\omega})$ определим множество $W_3(\overline{\omega}; v)$, состоящее из всех вершин u, удовлетворяющих условиям: $u < \omega^{(3)}(v)$, $u \in (U_4(\overline{\omega}; v) \setminus U_1(\mathbf{C}; v)) \bigcup V_0(\mathbf{C})$. Далее, для каждой вершины $v \in V_3(\overline{\omega})$ определим множество ребер $Y_3(\overline{\omega}; v)$, полагая $Y_3(\overline{\omega}; v) = \{\{u, v\} : u \in W_3(\overline{\omega}; v)\}$.

Введем обозначения: $Y_0(\mathbf{C}; \overline{\omega})$ — совокупность всех ребер из множества $X_{ad}(\mathbf{C})$, которые инцидентны хотя бы одной вершине из множества $V_0(\mathbf{C})$ и не являются инцидентными ни одной вершине из множества $V_3(\overline{\omega})$;

$$Y_1(\overline{\omega}) = \bigcup_{v \in V_1(\overline{\omega})} Y_1(\overline{\omega}; v); \quad Y_2(\overline{\omega}) = \bigcup_{v \in V_2(\overline{\omega})} Y_2(\overline{\omega}; v); \quad Y_3(\overline{\omega}) = \bigcup_{v \in V_3(\overline{\omega})} Y_3(\overline{\omega}; v);$$

$$X_{ad}(\mathbf{C}; \overline{\omega}) = Y_0(\mathbf{C}) \bigcup (Y_1(\overline{\omega}) \bigcup Y_2(\overline{\omega}) \bigcup Y_3(\overline{\omega})) \bigcap X_{ad}(\mathbf{C});$$

$\mathcal{B}_3(\mathbf{C}; \overline{\omega})$ — множество, состоящее из 3-блока $S(\mathbf{C}; \overline{\omega})$ и всех 3-блоков, получающихся добавлением к этому 3-блоку ребер из множества $X_{ad}(\mathbf{C}; \overline{\omega})$.

Оказалось, что для всякого 3-блока из множества $\mathcal{B}(\mathbf{C})$ можно указать такой триплет $\overline{\omega}(B) \in \Omega_0$, который удовлетворяет условию: блок B принадлежит множеству $\mathcal{B}_3(\mathbf{C}; \overline{\omega})$. Более того, множества $\mathcal{B}_3(\mathbf{C}; \overline{\omega})$ и $\mathcal{B}_3(\mathbf{C}; \overline{\omega}')$ не имеют общих 3-блоков, если $\overline{\omega}$ и $\overline{\omega}'$ — два различных триплета из множества $\Omega_0(\mathbf{C})$. Таким образом, множество 3-блоков $\mathcal{B}_3(\mathbf{C})$, где $\mathbf{C} \in \mathcal{C}(n)$, может быть представлено в виде разложения

$$\mathcal{B}_3(\mathbf{C}) = \bigcup_{\overline{\omega} \in \Omega_0(\mathbf{C})} \mathcal{B}_3(\mathbf{C}; \overline{\omega})$$

на непересекающиеся множества, то есть классы.

Литература

1. Калмыков Г.И. Каркасная классификация графов// В сб.: Труды IV Международной конференции "Дискретные модели в теории управляющих систем", Красновидово, 2000

(19–25 июня 2000 г.). Под ред. В.Б. Алексееева, В.А. Захарова. М.: МАКС Пресс, 2000, с. 34.

2. Калмыков Г. И.// Каркасная классификация помеченных блоков// В сб.: Материалы VII Международного семинара "Дискретная математика и ее приложения" (29 января–2 февраля 2001 г.). Часть II/ Под ред. О.Б. Лупанова. М.: Изд–во центра прикладных исследований при механико-математическом факультете МГУ, 2001, с. 221.

3. Калмыков Г.И. Представление вириальных коэффициентов, позволяющее избежать асимптотической катастрофы.// Теоретическая и математическая физика. – 2002. – Т. 130, N 3. – С. 508–528.

Оптимизация производственного плана по прибыли

А. В. Карповский, В. С. Рублев (Ярославль)

В современный период насущными становятся постановки задач оптимального планирования, где целью является не максимальный выпуск или минимальная себестоимость, а максимальная прибыль, которая может быть получена от реализации произведенного товара с учетом конъюнктуры рынка на текущий момент. При этом учитывается, что увеличение какого-либо вида товара на рынке приводит к снижению его рыночной цены и потому надо так распределить производство по разным видам товара, чтобы суммарная прибыль была максимальна.

Рассматривается следующая упрощенная модель задачи:

Имеется l видов сырья - $C_1, C_2, \ldots, C_l$, m видов оборудования - $E_1, E_2, \ldots, E_m$ и n видов производимой продукции - $P_1, P_2, \ldots, P_n$.

Введем обозначения: x_{ij} - план производства j-ой продукции на i-ом оборудовании ($x_{ij} \geq 0$), тогда $x_j = \sum\limits_{i=1}^{m} x_{ij}$ - план на j-ую продукцию в целом (т.е. по всем видам оборудования. Матрица плана $\{x_{ij}\}$ в данной задаче будет являться искомой). p_{ij} - время, затрачиваемое i-ым оборудованием на производство единицы j-ой продукции ($p_{ij} \geq 0$). T_i - ресурс времени эксплуатации i-го оборудования в плановый период ($T_i > 0$). c_k - объем k-го сырья, имеющийся на складах предприятия, с учетом возможности поступления сырья в плановый период ($c_k \geq 0$). c_{kj} - объем k-го сырья, затрачиваемый на производство единицы j-ой продукции ($c_{kj} \geq 0$).

Введем следующие ограничения. Во-первых, в плановый период мы не можем использовать оборудование дольше определенного времени, индивидуального для каждого вида:

$$\sum_{j=1}^{n} p_{ij} x_{ij} \leq T_i, \quad (i = \overline{1,m}). \tag{1}$$

Во-вторых, суммарное количество сырья, потраченное на производство всех видов продукции, не должно превышать объем, имеющийся на складах предприятия:

$$\sum_{j=1}^{n} c_{kj} x_j \leq c_k, \quad (k = \overline{1,l}). \tag{2}$$

Еще одно ограничение связано с зависимостью прибыли от количества продукции на рынке. Пусть функция зависимости прибыли от производства единицы j-ой продукции от

объема j-ой продукции имеет вид: $y_j = b_j - a_j \cdot x_j \, (a_j, b_j > 0)$. Тогда функция зависимости прибыли от производства всей j-ой продукции от объема j-ой продукции примет следующий вид: $f(x_j) = (b_j - a_j \cdot x_j)x_j$. Таким образом, максимум будет достигнут при $x_j = b_j/2a_j$ и третье ограничение примет вид:

$$\sum_{i=1}^{m} x_{ij} \leq \frac{b_j}{2a_j}, \quad (j = \overline{1,n}). \tag{3}$$

При других видах зависимостей цены от объема продаж, мы можем получить похожий вид кривой зависимости общей прибыли от производства j-ой продукции от объема произведенной j-ой продукции. В данном случае в "похожесть" вкладывается во-первых унимодальность функции, т.е. наличие одного максимума (в данном случае), а во вторых выпуклость функции (хотя это и не обязательно). Это внесет некоторые изменения в вид ограничения (3). Но на алгоритм проведения оптимизации это существенных изменеий не произведет. В дальнейшем, для определенности будем рассматривать именно линейную зависимость прибыли от объема производства.

При $f(x_j) = (b_j - a_j \cdot x_j) \cdot x_j$ целевая функция является квадратичной, а задача становится задачей квадратичного (выпуклого) программирования. В случае нелинейной зависимости цены от объема продаж, здесь берется точка максимума функции прибыли для данного вида продукции. В этом более общем случае, задача может перестать быть задачей выпуклого программирования, но все еще быть близкой к ней.

Для решения задачи предлагается следующий способ, основанный на методе градиентного спуска. Получившаяся в результате построения целевая функция, во-первых, будет вогнута, что следует из самого уравнения. Во-вторых, она будет унимодальной, т.е. функцией, которая имеет только один максимум. И, наконец, целевая функция может быть записана и как сумма линейной и квадратичной форм, т.е. данная задача относится к классу задач квадратичного программирования.

Алгоритм, использующийся в работе, основан на итерационном методе градиентного спуска с переменным шагом и выбран как наиболее простой в реализации и удобный для пользования. Суть алгоритма сводится к следующему: Положим $X^0 = x_{ij}^0$ - начальная матрица плана, удовлетворяющая всем предъявленным требованиям. Удобно в качестве начальной матрицы взять нулевую матрицу. Итерации проводятся следующим образом: $x_{ij}^s = x_{ij}^{s-1} + t \cdot (b_j - 2a_j \cdot x_{ij}^{s-1})$, $s = 1, 2, \ldots$ где в качестве t выбирается максимальное число, при котором выполнены неравенства условия задачи (ограничения). В случае табличной задачи зависимости прибыли от объема производства, итерации проводятся несколько иначе: $x_{ij}^s = x_{ij}^{s-1} + t \cdot \frac{f_j(x_j + \Delta x_j) - f_j(x_j)}{\Delta x_j}$, $s = 1, 2, \ldots$ Итерации проводятся до тех пор, пока не будет: $\left| \sum\limits_{j=1}^{n} f_j(x_j^s) - \sum\limits_{j=1}^{n} f_j(x_j^{s-1}) \right| < \varepsilon$ где в качестве ε можно взять $1/4$. Шаг t выбирается следующим образом:

$$t_1 \leq \min_{1 \leq i \leq m} \frac{T_i - \sum\limits_{j=1}^{n} p_{ij} x_{ij}^0}{\sum\limits_{j=1}^{n} p_{ij} \frac{b_j}{2a_j}}$$

$$t_2 \le \min_{1\le k\le l} \frac{c_k - \sum_{j=1}^{n} (c_{kj} \sum_{i=1}^{m} x_{ij}^0)}{\sum_{j=1}^{n} (c_{kj} \sum_{i=1}^{m} \frac{b_j}{2a_j})}$$

$$t_3 \le \min_{1\le i\le m} \frac{\frac{b_j}{2a_j} - \sum_{i=1}^{m} x_{ij}^0}{\sum_{i=1}^{m} \frac{b_j}{2a_j}}$$

$t = \frac{1}{2} \min\{t_1, t_2, t_3\}$

Полученный план при необходимости округляется до ближайших целых величин с проверкой допустимости. Поиск точного целочисленного решения практически не имеет смысла, так как те или иные отклонения в исходных данных плана (аварийный выход из строя оборудования, заболевание работников) ставят задачу по корректировке плана. При этом необходимо не пересчитывать план, а искать ближайшее оптимальное решение при поменявшихся данных. Эта задача является целью дальнейших исследований.

Проблемы разработки и применения итерационных алгоритмов для построения расписаний

В. А. Костенко (Москва)

В докладе будут рассмотрены:

1. основные особенности задач построения расписаний,
2. условия, ограничивающие применение для построения расписаний жадных алгоритмо в и алгоритмов, основанных на частичном переборе,
3. проблемы и пути их решения [1,2] при разработке итерационных алгоритмов построения расписаний: алгоритмы детерминированной коррекции расписаний, генетические и эволюционные алгоритмы, алгоритмы имитации отжига, алгоритмы случайного поиска (ненаправленного, направленного, направленного с самообучением).

Литература

1. Костенко В.А. Задача построения расписания при совместном проектировании аппаратных и программных средств// Программирование, 2002., N 3, С.64-80.

2. Костенко В.А. Проблемы разработки итерационных алгоритмов для построения расписаний с одновременным нахождением необходимого количества ресурсов и их характеристик// Искусственный интеллект (Донецк), 2002., N 2, С.141-150.

О восстановлении автоматов по траекториям в пространстве состояний

В. А. Козловский, Л. А. Толмачевская (Донецк)

В ряде задач идентификации моделей динамических объектов основной информацией является информация о текущем состоянии объекта, которая обеспечивается наблюдением в пространстве состояний объекта в течение всего интервала идентификации [1]. По

этим данным восстанавливаются основные параметры системы, характеризирующие ее поведение.

В данной работе рассматривается подобная задача в автоматной постановке, а именно по наблюдаемой последовательности состояний некоторого автомата из заданного класса восстановить его таблицу с определенной точностью.

В традиционной теории экспериментов с автоматами и ее обобщениях [2] разработаны подходы к решению аналогичной задачи по наблюдаемому поведению автомата. Опишем один из возможных вариантов сведения задачи восстановления автоматов по траектории в пространстве состояний к традиционной на основе определения так называемого двойственного автомата.

Пусть $A = (S_A, X_A, Y_A, \delta_A, \lambda_A)$, где S_A - множество состояний (внутренний алфавит автомата), X_A - входной алфавит, Y_A - выходной алфавит, $\delta_A : S_A \times X_A \to S_A$ - функция переходов, $\lambda_A : S_A \times X_A \to Y_A$ - функция выходов.

Под гомоморфизмом автомата $A = (S_A, X_A, Y_A, \delta_A, \lambda_A)$ на автомат $B = (S_B, X_B, Y_B, \delta_B, \lambda_B)$ понимается тройка отображений (φ, ψ, θ) $\varphi : S_A \to S_B$, $\psi : X_A \to X_B$, $\theta : Y_A \to Y_B$ таких, что $\varphi(\delta_A(s, x)) = \delta_B(\varphi(s), \psi(x))$, а $\theta(\lambda_A(s, x)) = \lambda_B(\varphi(s), \psi(x))$. Гомоморфизм назовем гомоморфизмом по внутреннему алфавиту, если φ является тождественным отображением, и гомоморфизмом по внешнему алфавиту, если отображения ψ и θ - тождественные.

В [2] изучались фрагменты и представления автоматов, которые будем называть фрагментами и представлениями по внешнему алфавиту. Определим фрагменты и представления автоматов по внутреннему алфавиту.

Пусть IR - частичный автомат, множество состояний которого совпадает с множеством состояний автомата A. Он называется фрагментом по внутреннему алфавиту автомата A, если существует гомоморфизм по внутреннему алфавиту автомата OR в A.

Последовательность переходов из состояния в состояние ("траектория в пространтстве состояний автомата") является частным случаем фрагментов по внутреннему алфавиту автомата.

Пусть F - класс автоматов, имеющих одинаковые множества состояний S.

Представления будем рассматривать с точностью до изоморфизма (точность может быть и иной).

Представлением по внутреннему алфавиту автомата A относительно класса F называется фрагмент OR этого автомата, из гомоморфизма которого на автомат класса F следует изоморфизм этого автомата автомату A.

Двойственным к автомату $A = (S_A, X_A, X_A, \delta_A, \lambda_A)$ назовем автомат $A^\Delta = (S_{A^\Delta}, X_{A^\Delta}, X_{A^\Delta}, \delta_{A^\Delta}, \lambda_{A^\Delta})$, где $S_{A^\Delta} = X_A$, $X_{A^\Delta} = S_A$, $\delta_{A^\Delta}(x, y) = \lambda_A(y, x)$, $\lambda_{A^\Delta} = \delta_A(y, x)$, дл любых $y \in S_A$, $x \in X_A$.

Очевидно, что двойственным для двойственного автомата является исходный $(A^\Delta)^\Delta = A$.

Классу F автоматов поставим в соответствие двойственный класс F^Δ, дл каждого автомата которого существует двойственный автомат в классе F.

Далее полагаем, что у автомата A совпадают входной и выходной алфавиты, тогда дл него существует двойственный.

Теорема 1. *Фрагмент R по внешнему алфавиту автомата A является представлением этого автомата относительно класса F тогда и только тогда, когда двойственный к*

нему автомат R^Δ является представлением по внутреннему алфавиту автомата A^Δ, двойственного к автомату A, относительно класса F^Δ, двойственного классу F.

Для некоторых классов автоматов это утверждение позволяет перенести результаты, полученные для представлений по внешнему алфавиту на представления по внутреннему алфавиту.

В частности для группового автомата двойственным является автомат без потери информации порядка 1 (БПИ-1) [3] и обратно.

Пусть A – БПИ-1 автомат, граф переходов которого не является мультиграфом, принадлежащий классу F всех БПИ-1 автоматов с таким же множеством состояний, что и A, с одинаковыми входными и выходными алфавитами, мощности которых совпадают с мощностью соответствующих алфавитов автомата A.

Пусть предъявляется фрагмент IR по внутреннему алфавиту в пространстве состояний этого автомата A. Ставится задача распознавания свойства "быть представлением по внутреннему алфавиту" автомата A относительно класса F. Эту задачу назовем задачей распознавания представления по внутреннему алфавиту.

Используя результаты полученные для групповых автоматов в [4] можно показать следующее.

Теорема 2. *Задача распознавания представления по внутреннему алфавиту при указанных выше предположениях является NP-полной.*

Литература

1. Панасенко В.В., Соколов С.В., Щербань И.В. Решение задачи идентификации модели динамического объекта при однократном наблюдении его состояния. // Известия Академии наук. Теория и системы управления, 2003, 1, с. 24-28.

2. Грунский И.С., Козловский В.А., Пономаренко Г.Г. Представления конечных автоматов фрагментами поведения – Киев: Наукова думка, 1990. – 230 с.

3. Гилл А. Введение в теорию конечных автоматов – М.: Наука, 1966. – 272 с.

4. Козловский В.А. О представлениях групповых автоматов // Кибернетика и системный анализ, 1996, 2, с. 21-28.

Вероятностный анализ жадного алгоритма для задачи о покрытии

Н. Н. Кузюрин (Москва)

Рассматривается задача о покрытии, которая записывается в виде задачи ЦЛП следующим образом:

$$\{\min \mathbf{cx} |\ A\mathbf{x} \geq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{x} - \text{целочисленный}\},$$

где $\mathbf{c} = (1, ..., 1)$, $\mathbf{b} = (1, ..., 1)^T$ и $A = (a_{ij})$ произвольная $m \times n$ матрица, $a_{ij} \in \{0, 1\}$ для всех $i,\ j$.

Одним из самых простых эвристических алгоритмов решения задачи о покрытии является жадный (градиентный) алгоритм. Обозначим через $C_G(A)$ – размер покрытия, получаемого с помощью жадного алгоритма, через $C(A)$ – размер минимального покрытия

и положим

$$R(A) = \frac{C_G(A)}{C(A)}.$$

Известно, что $R(A) \leq 1 + \ln m$ ([1-3]). Недавно в [4] эта оценка была немного уточнена и доказано, что $\max_A R(A) = \ln m - \ln\ln m + \theta(1)$ (максимум берется по всем матрицам A с m строками). С другой стороны, в [5] доказано, что в задаче о покрытии, никакой полиномиальный алгоритм не может гарантировать мультипликативную точность $(1-\delta)\ln m$ (для любого фиксированного $\delta > 0$), если только не выполнено $NP \subseteq DTIME[n^{O(\log\log n)}]$.

Однако, все эти результаты относятся к анализу по худшему случаю. Мы рассматриваем ситуацию, когда все элементы матрицы A являются независимыми одинаково распределенными случайными величинами, причем $\mathbf{P}\{a_{ij} = 1\} = p$, $\mathbf{P}\{a_{ij} = 0\} = 1 - p$, и исследуем точность жадного алгоритма для таких случайных данных. Значение $R(A)$ при этом является случайной величиной.

Теорема. *Пусть p таково, что $p \leq c < 1$ для произвольной константы c и*

$$\frac{\ln\ln np}{\ln mp} \to 0 \ \textit{при}\ n \to \infty, \tag{12}$$

$$\frac{\ln m}{np} \to 0 \ \textit{при}\ n \to \infty. \tag{13}$$

Тогда для любого фиксированного $\varepsilon > 0$ $\mathbf{P}\{R(A) \leq 1 + \varepsilon\} \to 1$ при $n \to \infty$.

Это означает, что на случайных данных жадный алгоритм является $(1+\varepsilon)$-приближенным алгоритмом. Работа выполнена при поддержке РФФИ, проект 02-01-00713.

Литература

1. Нигматуллин Р.Г., Метод наискорейшего спуска в задачах на покрытие, в сб: Вопросы точности и эффективности алгоритмов (труды симпозиума), вып. 5, Киев, 1969, с. 116-126.

2. Сапоженко А.А., О сложности днф, получаемой с помощью градиентного алгоритма, Дискретный анализ, вып. 21, Новосибирск, 1972, с. 62-71.

3. Lovasz L., On the ratio of optimal integral and fractional covers, Discrete Math. **13** (1975) 383–390.

4. Slavik P., A tight analysis of the greedy algorithm for set cover, J. of Algorithms, **25** (1997) 237–254.

5. Feige U., A threshold of $\ln n$ for the approximating set cover, Proceedings of the Annual ACM Symposium on Theory of Computing, (1996) 314–318.

Корректирующая способность двоичных линейных кодов и монотонные функции

В. И. Левенштейн (Москва)

Пусть F^n - множество всех двоичных векторов $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ (с координатами 0 и 1) и F_t^n - множество всех векторов F^n, которые имеют число единиц (вес) t. Для любого $\mathbf{x} \in F^n$ рассмотрим множество $S(\mathbf{x}) = \{i : x_i = 1\}$ и определим на F^n частичный порядок $\subseteq$

(покрытие): $\mathbf{x} \subseteq \mathbf{y}$ тогда и только тогда $S(\mathbf{x}) \subseteq S(\mathbf{y})$. Линейный код $C \subseteq F^n$ размерности k будем называть $[n,k]$ кодом. Мы обозначаем d_C и r_C минимальное расстояние и радиус покрытия кода C соответственно. *Лидер* смежного класса $[n,k]$ кода C - это лексикографически наименьший элемент среди векторов наименьшего веса в этом смежном классе. Обозначим $E^0(C)$ множество лидеров всех 2^{n-k} смежных классов. При декодировании по минимальному расстоянию (см. [1]) только векторы ошибок из множества $E^0(C)$ могут быть исправлены и они все исправляются при передаче любого кодового вектора. Поэтому элементы $E^0(C)$ называются *исправимыми ошибками*, а элементы $E^1(C) = F^n \setminus E^0(C)$ - *неисправимыми ошибками*. Пусть $E_t^0(C) = E^0(C) \cap F_t^n$ - множество исправимых ошибок веса t и $E_t^1(C) = E^1(C) \cap F_t^n$ - множество неисправимых ошибок веса t. Важным свойством линейных кодов является монотонная структура множеств исправимых и неисправимых ошибок (см. [1]). Это свойство может быть сформулировано следующим образом.

Лемма 1. *Для любого $[n,k]$ кода C булева функция $f_C(\mathbf{x})$ такая, что $f_C(\mathbf{x}) = 1$, если $\mathbf{x} \in E^1(C)$, и $f_C(\mathbf{x}) = 0$, если $\mathbf{x} \in E^0(C)$, является монотонной.*

Одной из основных задач теории кодирования является нахождение для $[n,k]$ кода C функции $\varepsilon_C(t) = |E_t^0(C)|/\binom{n}{t}$, которая характеризует способность кода исправлять ошибки веса t, $t = 0, 1, ..., n$. По лемме 1 исследование корректирующей способности кодов C связано с исследованием соответствующих монотонных функций $f_C(\mathbf{x})$. В частности, из леммы 1 следует, что для любого $[n,k]$ кода C и любого $t = 0, 1, ..., n-1$, имеет место $\varepsilon_C(t+1) \leq \varepsilon_C(t)$.

Обозначим $M^1(C)$ множество минимальных неисправимых ошибок, т.е. векторов $\mathbf{y} \in E^1(C)$ таких, что если $\mathbf{x} \subseteq \mathbf{y}$ и $\mathbf{x} \in E^1(C)$, то $\mathbf{x} = \mathbf{y}$ (множество нижних единиц функции $f_C(\mathbf{x})$). Вектор $\mathbf{u} \in F^n$ будем называть *"большой половинкой"* кодового слова $\mathbf{c} \in C$, $\mathbf{c} \neq \mathbf{0}$, тогда и только тогда, когда $\mathbf{u} \subseteq \mathbf{c}$, $||\mathbf{c}|| \leq 2||\mathbf{u}|| \leq ||\mathbf{c}|| + 2; m(\mathbf{u}) = m(\mathbf{c})$, если $2||\mathbf{u}|| = ||\mathbf{c}||; m(\mathbf{u}) > m(\mathbf{c})$, если $2||\mathbf{u}|| = ||\mathbf{c}|| + 2$ (здесь $m(\mathbf{x}) = \min S(\mathbf{x})$). Вектор $\mathbf{c} \in C$, $\mathbf{c} \neq \mathbf{0}$, называется *минимальным*, если $\mathbf{a} \subset \mathbf{c}$, где $\mathbf{a} \in C$, влечет $\mathbf{a} = \mathbf{0}$.

Теорема 1. *Для любого $[n,k]$ кода C множество $M^1(C)$ состоит из больших половинок минимальных кодовых векторов.*

Градиентный метод декодирования по минимальному расстоянию основан на использовании проверочных множеств для $[n,k]$ кода C. Множество $T \subseteq C$ называется *проверочным* [2], если для любого $\mathbf{x} \in E^1(C)$ существует вектор $\mathbf{c} \in T$ такой, что вес $\mathbf{x} + \mathbf{c}$ меньше веса $\mathbf{x}$ или веса этих векторов равны и $m(\mathbf{c}) = m(\mathbf{x})$ (т.е. $\mathbf{x}+\mathbf{c}$ лексикографически меньше, чем $\mathbf{x}$). Теорема 1 позволяет строить проверочные множества, состоящие из меньшего числа слов, по сравнению с известным проверочным множеством, образованным всеми минимальными векторами кода [2]. Это утверждение позволяет также усилить известные оценки для $\varepsilon_C(t)$ и получить ряд новых оценок.

Теорема 2. *Для любого $[n,k]$ кода C и любого целого t, $d_C/2 \leq t \leq r_C$,*

$$\varepsilon_C(t+1) \leq \varepsilon_C(t) - 1/\binom{n}{t}.$$

Доказательство основано на том факте, что для любого t, $d_C/2 \leq t \leq r_C$, существует по крайней мере $n - t + 1$ пар $(\mathbf{x}, \mathbf{y})$ таких, что $\mathbf{x} \in \mathbf{E_{t-1}^0(C)}$ и $\mathbf{y} \in \mathbf{E_t^1(C)}$.

В настоящей работе (которая является частью исследования [3]) получены также простые и точные границы для $\varepsilon_C(t)$ в терминах величины $\sigma(n,k,t) = 2^{k-n} \sum_{i=0}^{t} \binom{n}{i}$. Заметим,

что $\sigma(n,k,t) \leq 1$ является необходимым условием для существования $[n,k]$ кода с исправлением t ошибок (граница Хэмминга).

Теорема 3. *Для любого* $[n,k]$ *кода* C *и любого* t, $t = 0,1,...,n$,

$$\varepsilon_C(t) \leq \frac{2^{n-k}}{\sum_{i=0}^{t} \binom{n}{i}} = \frac{1}{\sigma(n,k,t)}.$$

Для любых $n,k,$ *и* $t,$ $0 \leq t \leq n$, *существует* $[n,k]$ *код* C *такой, что*

$$\varepsilon_C(t) > 1 - \sigma(n,k,t).$$

Асимптотические следствия для последовательности $[n,k]$ кодов C и весов t при $n \to \infty$ основаны на следующих соображениях. Если для такой последовательности имеет место $\sigma(n,k,t) \to \infty$, то $\varepsilon_C(t) \to 0$. С другой стороны, если параметры k,n,t удовлетворяют условию $\sigma(n,k,t) \to 0$, то существуют последовательность $[n,k]$ кодов C таких, что $\varepsilon_C(t) \to 1$.

Работа выполнена при финансовой поддержке гранта РФФИ 01-01-00035.

Литература

1. У. Питерсон, Э. Уэлдон, Коды, исправляющие ошибки, Мир, Москва, 1976.

2. A. Ashikhmin, A. Barg, Minimal vectors in linear codes, IEEE Trans. on Inform. Theory, 1998, vol. 44, pp. 2010–2017.

3. T. Helleseth, T. Kløve, V.I. Levenshtein, Error-correction capability of binary linear codes and the discrete simplex problem, submitted to IEEE Trans. on Inform. Theory.

Взаимодействия динамической информационной модели DIM

Д. И. Лобачев, В. С. Рублев (Ярославль)

В [1] отмечена ключевая роль введения взаимодействий объектной системы DIM (см. описание в [2]), которые соответствуют методам классов отношений объектов. С помощью них описываются любые действия, которые могут произойти с объектом, группой объектов, а также действия, инициированные внешним миром по отношению к информационной системе. Взаимодействие несет в себе всю необходимую информацию, требуемую для того или иного действия. В отличие от объектно-ориентированных баз данных, где связь между объектами служит для их расстановки в некоторой иерархии наследования и поддержания целостности базы данных, в системе DIM взаимодействие позволяет также указать количественные характеристики, наложить условия и ограничения для начала выполнения взаимодействия между двумя и более объектами, проинициализировать и возбудить дальнейшие взаимодействия.

В общем случае в системе DIM выделяется три типа взаимодействия, которыми можно описать любое действие в информационной системе.

К первому типу относятся взаимодействия пользователя с системой. При этом пользователь может *создавать* новые объекты и их связи с другими объектами, *корректировать*

данные объектов и связей или *удалять* их. Данный вид взаимодействия тесно интегрирован с клиентской частью системы DIM по построению динамических форм (см. [3]). Взаимодействие несет в себе информацию об уровнях доступа к тому или иному классу или объекту, хранит специфические процедуры, запускающиеся до или после определенных действий пользователя.

Ко второму типу относятся взаимодействия, связанные с физическим движением материальных объектов из одних *подразделений-объектов* в другие. Каждому материальному объекту соответствует информационный объект, для которого определяются его ресурс в подразделении и лимиты ресурса. При выполнении взаимодействия изменяются ресурсы информационного объекта в объектах-подразделениях и проверяются верхние и нижние границы ресурсов. При нарушении ограничений возбуждаются события требующие выполнения взаимодействий по восстановлению ресурсов в требуемых границах.

Третий тип взаимодействия определяет технологическую операцию изготовления продукции из других видов продукции. Во время выполнения взаимодействия выполняются процедуры по исследованию наличия у подразделений *ресурсов*, необходимых для изготовления заданного объема продукции, разрешение на их использование, возбуждаются события, связанные с нарушениями лимитов. С помощью третьего типа взаимодействия, при наличии всей необходимой информации в информационной системе, можно *виртуально* запустить производство и получить различные варианты исполнения плана.

Выделение этих основных типов взаимодействий должно способствовать значительному уменьшению кода процедур, реализующих действия в сложных информационных системах, связанных с производством.

Литература

1. Рублев В.С., Дерябин В.О., Иоссель М.А., Карповский А.В., Лобачев Д.И., Юсупов А.Р. Классы отношений объектов и взаимодействия объектов // Дискретные модели в теории управляющих систем. Тезисы докладов V научной конференции (Дубна, 26-29 мая 2003 г.)

2. Дерябин В. О., Лобачев Д. И., Рублев В. С., Юсупов А. Р. Базовые отношения объектов динамической информационной модели и гибкие таблицы данных // Проблемы теоретической кибернетики. Тезисы докладов XIII Международной конференции (Казань, 27–31 мая 2002 г.). Часть I. — М.:изд-во механико-математического факультета МГУ, 2002. — С.55.

3. Юсупов А.Р. Интерфейс пользователя динамической информационной модели DIM и навигатор объектов // Дискретные модели в теории управляющих систем. Тезисы докладов V научной конференции (Дубна, 26-29 мая 2003 г.)

О структуре минимальных схем из функциональных элементов в базисе $\{\&, \vee, \neg\}$, реализующих линейную функцию

С. А. Ложкин (Москва)

В [1] было установлено, что при $n \geq 2$ сложность реализации любой из линейных функций $x_1 \oplus \ldots \oplus x_n$ или $x_1 \oplus \ldots \oplus x_n \oplus 1$ схемами из функциональных элементов (СФЭ) в

базисе $\{\&, \vee, \neg\}$ равна $(4n-4)$. Нетрудно убедиться в том, что в базисе $\{\&, \vee, \neg\}$ имеется единственная (с точностью до изоморфизма) минимальная СФЭ, – макроэлемент типа 0, – реализующая функцию $x_1 \oplus x_2$, и единственная минимальная СФЭ, – макроэлемент типа 1, – реализующая функцию $x_1 \oplus x_2 \oplus 1$. При этом макроэлемент типа 0 (типа 1) представляет собой формулу $(x_1 \vee x_2) \cdot \overline{x_1 \cdot x_2}$ (соответственно $x_1 \cdot x_2 \vee \overline{x_1 \vee x_2}$). В настоящей работе доказано, что любая минимальная СФЭ в базисе $\{\&, \vee, \neg\}$, реализующая любую из функций $x_1 \oplus \ldots \oplus x_n$, $x_1 \oplus \ldots \oplus x_n \oplus 1$, состоит из $(n-1)$ макроэлементов и представляет собой их бесповторную суперпозицию.

Работа выполнена при финансовой поддержке РФФИ, грант 02-0101110.

Литература

1. Редькин Н.П. Доказательство минимальности некоторых схем из функциональных элементов // Проблемы кибернетики, 1970, вып.23, с.83-102

Суффиксный метод получения нижних оценок сложности схем композиции слов

Ю. В. Мерекин (Новосибирск)

В работе исследуется сложность процедуры построения слов, когда разрешается многократное использование уже построенных слов. Для построения слов в алфавите $\Sigma = \{a_0, \ldots, a_{q-1}\}$, $q \geq 1$, используем операцию композиции, впервые предложенную А. И. Ширшовым [3] в 1962 году. Пусть U_1, V_1, R — произвольные слова (возможно, R пусто) и $U = U_1R$, $V = RV_1$. Операция *композиции* слов U и V относительно слова R определяется как слово U_1RV_1 и обозначается через $(U \bullet V)_R = U_1RV_1$. В некоторых случаях знак $\bullet$ опускается. При пустом R это не что иное как *конкатенация* $U \bullet V$.

Последовательность слов $a_0, a_1, \ldots, a_{q-1}, X, Y, \ldots, Z$ называется *схемой композиции* слова Z и обозначается через S, если для любого слова W из этой последовательности, начиная со слова X, в S имеются такие слова $U = U_1R$, $V = RV_1$ (возможно, R пусто, $U = V$), предшествующие слову W, что $W = (U \bullet V)_R$. Если для построения всех слов $X, Y, \ldots, Z$ применяется только операция конкатенации, то схема имеет специальное название — *схема конкатенации* слова Z.

Под *сложностью* $L_{sh}(S)$ схемы S композиции слова Z понимается число слов в последовательности $X, Y, \ldots, Z$. Пусть $L_{sh}(Z) = \min L_{sh}(S)$, где минимум берется по всевозможным схемам композиции слова Z. Величину $L_{sh}(Z)$ назовем *мультипликативной сложностью* слова Z. Как и ранее $L(Z)$ обозначает сложность слова Z для схем конкатенации.

Слово V называется *подсловом* слова W и обозначается через $V \sqsubseteq W$, если для некоторых (возможно, пустых) слов X и Y справедливо равенство $W = XVY$. При пустом X подслово V называется *префиксом*, а припустом Y — *суффиксом* слова W. При получении нижних оценок мультипликативной сложности слов в классе схем композиции слов применим суффиксный метод, предложенный в [1] для схем конкатенации слов и позволивший получить нижние оценки сложности реализации некоторых классов слов (в частности, определяемых симметрическими булевыми функциями [2]).

Суффиксный метод использует специальные представления слов. Пусть слово W представлено в виде $W = UxV$, где $x \in \Sigma$, и, возможно, U пусто. Если V является либо символом, отсутствующим в слове Ux, либо $V \sqsubseteq Ux$ и $xV \not\sqsubseteq U$, то слово V называется *максимальным суффиксом* слова W (однобуквенное слово по определению является своим максимальным суффиксом). Представление слова W в виде $W = X_1 \bullet \ldots \bullet X_r$ называется *суффиксным представлением* , если длина слова X_1 равна единице, а каждое слово X_i, $1 \le i \le r$, является максимальным суффиксом слова $X_1 \bullet \ldots \bullet X_i$. Очевидно, что суффиксное представление любого слова единственно. Число операций конкатенации в суффиксном представлении слова W называется *суффиксной сложностью* слова W и обозначается через $L^*(W)$. Следующая теорема устанавливает соотношения между $L(W)$, $L_{sh}(W)$ и $L^*(W)$.

Теорема. *Для всякого слова W выполняются неравенства $L(W) \ge L_{sh}(W) \ge L^*(W)$.*

Базис операций схем композиции строго включает в себя базис операций схем конкатенации. Поэтому для всякого слова W очевидно неравенство $L(W) \ge L_{sh}(W)$. В [1] получено неравенство $L(W) \ge L^*(W)$. В теореме при доказательстве справедливости неравенства $L_{sh}(W) \ge L^*(W)$ используются приемы из предыдущих работ автора. При этом ключевую роль играет обобщение свойства соотношения слов в заключительной операции схемы композиции слова W.

Расширение возможностей базисной операции при переходе от схем конкатенации слов к схемам композиции иногда позволяет получить более эффективную оценку сложности. В качестве примера приведем задачу построения слов в однобуквенном алфавите $\Sigma = \{a\}$. В настоящее время точное решение этой задачи в классе схем конкатенации слов не получено. В классе схем композиции слов при любом натуральном n для всех слов a^t, $2^{n-1} < t \le 2^n$, справедливо равенство $L_{sh}(a^t) = n$, а для построения набора слов $a^{t_1}, \ldots, a^{t_m}$, $2^{n-1} < t_1 < \ldots < t_m \le 2^n$, имеем $L_{sh}(a^{t_1}, \ldots, a^{t_m}) = n + m - 1$.

Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 02-01-00939).

Литература

1. Мерекин Ю. В. Нижняя оценка сложности для схем конкатенации слов // Дискрет. анализ и исслед. операций. 1996. Т. 3, N 1. С. 52–56.

2. Мерекин Ю. В. Нижние оценки сложности символьных последовательностей, определяемых симметрическими булевыми функциями // Дискрет. анализ и исслед. операций. Сер. 1. 2000. Т. 7, N 2. С. 54–64.

3. Ширшов А. И. Некоторые алгоритмические проблемы для алгебр Ли // Сиб. мат. журн. 1962. Т. 3, N 2. С. 292–296.

К вопросу описания решетки замкнутых классов конечнозначной логики

Е. А. Михеева (Ульяновск)

Проблема описания решетки (по включению) L_k всех замкнутых классов k-значной логики P_k решена полностью лишь в случае $k = 2$ [1]. Изучение L_k при $k > 2$ наталкивается на значительные трудности, поскольку в P_k есть классы, не имеющие конечных базисов,

и L_k имеет мощность континуума [2]. Как известно [2,3], в L_k континуум дают как классы со счетным базисом, так и классы без всякого базиса. Семейство всех замкнутых классов из P_k, не имеющих конечных базисов, обозначим через B_k. В семействе B_k выделим максимальные классы, т.е. такие классы, которые сами не имеют конечных базисов, а все их собственные замкнутые надклассы конечными базисами обладают.

На взгляд автора данной работы описание решетки L_k следует вести в двух направлениях:

1) Исследование структуры классов из B_k для выработки критерия распознавания полноты бесконечных систем.

2) Описание максимальных классов семейства B_k для выработки критерия распознавания полноты конечных систем.

Кратко можно выделить следующие наиболее важные результаты, полученные в этих направлениях.

1. Дана классификация нижних окрестностей замкнутых классов из решетки L_k [3]. Показаны некоторые особенности нижней окрестности классов из B_k.

Определение. Цепь $U_0 \subset U_1 \subset ... \subset U_n \subset ...$, пределом которой является класс U, называется особой в U, если найдется n_0 такое, что класс U_n не содержится ни в каком предполном в U классе для каждого $n \geq n_0$.

Из работы [4] вытекает, что класс с конечным базисом не является пределом ни какой строго возрастающей цепи замкнутых классов, и, следовательно, он не имеет особых цепей.

Доказана

Теорема 1. *По числу содержащихся в нижней окрестности предполных классов и максимальных классов особых цепей все классы из L_k подразделяются на девять типов:*

$$(0,0);(0,n);(0,\infty);(n,0);(n,n);(n,\infty);(\infty,0);(\infty,n);(\infty,\infty),$$

где n - конечное число, большее нуля; ∞ - бесконечное число; первый элемент пары означает число предполных классов, второй - число максимальных классов особых цепей.

Замечание 1. Классы типов $(0,0)$ и $(n,0)$ являются классами с конечным базисом, классы остальных типов составляют семейство B_k.

2. Доказаны в [5]

Теорема 2. *В P_k при $k > 2$ максимальные классы существуют, причем их множество не более чем счетно, и каждый класс из B_k содержится в некотором максимальном.*

Замечание 2. Максимальные классы в L_k интересны тем, что с одной стороны, они содержат в себе все классы из B_k и, с другой стороны, вне них в решетке L_k лежат только классы с конечным базисом.

Теорема 3. *В P_k для каждого $k > 2$ построен максимальных класс, имеющий в L_k глубину 5 при $k = 3$ и глубину 3 при $k > 3$.*

Замечание 3. Из вышеуказанных результатов вытекает достаточное условие конечной порожденности: замкнутый класс, не содержащийся целиком ни в одном из максимальных классов, обладает конечным базисом.

Работа выполнена в рамках программы "Университеты России" (код УР.04.01.035).

Литература

1. Post E. Two-valued iterative systems of mathematical logic. // Ann. Math. Studies, 1941, v.5.

2. Янов Ю.И., Мучник А. А. О существовании k-значных замкнутых классов, не имеющих конечного базиса. // ДАН СССР (1959) 127, N1, с.44-46.

3. Михеева Е.А. Классификация нижних окрестностей замкнутых классов из решетки L_k. // Дискретная математика, 1991, Т.3, вып. 4, с.3-15.

4. Яблонский С.В. Функциональные построения в k-значной логике. // Труды МИАН, 1958, Т.51, с.5-142.

5. Михеева Е.А. Построение в P_k максимальных классов, не имеющих конечных базисов // Дискретная математика, 1998, т.10, вып.2, с.137-159.

Сложность сортировки k-значного n-мерного куба

А. С. Нагорный (Москва)

Объектом исследования данной работы является задача сортировки частично упорядоченных множеств (ЧУМ). Приведем необходимые определения.

Пусть $\{X_N\}$ — произвольная бесконечная последовательность ЧУМ, где $|X_N| = N$, $N \to \infty$. Решить задачу сортировки $\{X_N\}$ означает найти алгоритм, который, получая на входе последовательность $x_1, x_2, \ldots, x_N$ элементов произвольного ЧУМ $X_N \in \{X_N\}$, выполняет их идентификацию (с точностью до автоморфизмов ЧУМ X_N). Таким образом, на выходе этого алгоритма будет некоторая "расстановка" входных элементов в соответствии с заданным отношением частичного порядка ЧУМ X_N.

В качестве элементарного шага сортирующего алгоритма рассматривается выполнение операции сравнения между собой любых двух различных элементов x_i и x_j входной последовательности. Далее, под сложностью алгоритма, сортирующего X_N, понимается количество попарных сравнений элементов, достаточное для выполнения идентификации всех входных элементов $x_1, x_2, \ldots, x_N$ при произвольном их начальном расположении.

Наконец, в качестве меры сложности задачи сортировки последовательности ЧУМ $\{X_N\}$ берется сложность оптимального (т.е. самого быстрого) алгоритма, сортирующего любое ЧУМ X_N из $\{X_N\}$.

В работе [1] В.В.Морозенко был рассмотрен случай $X_N = B^n$ (B^n — n-мерный булев куб, $n = \log_2 N$). Для сложности задачи сортировки B^n было получено асимптотическое равенство

$$L(B^n) \sim N \log_2 N, \quad n \to \infty.$$

Автору этой работы ранее удалось доказать для более общего случая $X_N = E_k^n$ (т.е. X_N – k-значный n-мерный куб) аналогичную нижнюю оценку (см. [2]).

В данном докладе приводится алгоритм, сортирующий E_k^n, за время, "близкое" к оптимальному. А именно, основной результат, полученный в этой работе, следующий.

Теорема. *Существует алгоритм A, сортирующий k-значный n-мерный куб, и имеющий сложность $L(A)$, удовлетворяющую следующему асимптотическому неравенству*

$$L(A) \lesssim \lceil \log_2 k \rceil \cdot N \log_k N, \quad n \to \infty.$$

(здесь $n = \log_k N$).

Заметим, что если k есть степень двойки, то полученные верхняя и нижняя оценки асимптотически совпадают.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 03-01-00783).

Литература

1. В.В.Морозенко. О сложности сортировки булевой алгебры, "Дискретная Математика", т.3, вып.1, 1991, стр.42–47.

2. А.С.Нагорный. Сложность сортировки n-й декартовой степени частично упорядоченного множества, "Проблемы теоретической кибернетики. Тезисы докладов XI Международной конференции 10-14 июня 1996 г.

О числе и структуре множеств, свободных от сумм в отрезке натуральных чисел

К. Г. Омельянов, А. А. Сапоженко (Москва)

Подмножество A целых чисел называется *свободным от сумм*, если для любых $a, b \in A$ число $a+b$ не принадлежит множеству A. Для любых действительных чисел q и p обозначим через $[q,p]$ множество натуральных чисел x таких, что $q \le x \le p$. Семейство всех подмножеств $A \subseteq [t,n]$, свободных от сумм, обозначим через $S(t,n)$. Пусть $s(t,n) = |S(t,n)|$, а $s(n) = |S(1,n)|$. П. Камерон и П.Эрдеш [4] предположили, что $s(n) = O(2^{n/2})$. Кроме того в [1] авторы получили оценку $s(n/3,n) = O(2^{n/2})$. Н. Калкин [3] и независимо Н. Алон [2] доказали, что $s(n) = 2^{n(1/2+o(1))}$. В [1] доказано, что для всякого $\varepsilon > 0$ существует константа $c = c(\varepsilon)$, такая, что $s((1/4+\varepsilon)n, n) \le c2^{n/2}$. Основным результатом является следующая

Теорема 1. *Пусть $t \ge n^{3/4}\log_2 n$. Тогда*

$$s(t,n) = O(2^{n/2}). \tag{14}$$

Это утверждение вытекает из теоремы 2, описывающей строение множеств $A \in S(1,n)$, содержащих "достаточно плотный" отрезок "небольшой" длины. Из теоремы 2 вытекает также независимое доказательство упомянутого выше результата Н.Алона и Н.Калкина.

Теорема 2. *Пусть $\hat{q} = n^{3/4}\log n$, $\varepsilon_n = \log\log n/\sqrt{\log n}$ и $\omega_n = \sqrt{n}\log\log n$. Тогда*

$$\hat{s}(q,n) \le \begin{cases} 2^{n/2-q/4+O(q\varepsilon_n)} & \text{при} \quad \hat{q} \le q \le 4n/9, \\ 2^{q+O(n^{3/4}\sqrt{\log n})} & \text{при} \quad \frac{4n}{9} < q < \frac{n}{2} - \hat{q}, \\ 2^{(n-q)+O(n\varepsilon_n+\omega_n)} & \text{при} \quad n/2+\hat{q} \le q \le n. \end{cases} \tag{15}$$

Кроме того, существуют абсолютные константы $0 < c_1 \le c_2$, такие, что

$$c_1 2^{n/2} \le \sum_{n/2-\hat{q} \le q \le n/2+\hat{q}} \hat{s}(q,n) \le c_2 2^{n/2}. \tag{16}$$

Работа выполнена при поддержке РФФИ (проект 01-01-00266).

Литература

1. Омельянов К. Г., Сапоженко А. А., О числе множеств, свободных от сумм, в отрезке натуральных чисел.// Дискретная математика (2002) 14, No 3, 2002, с. 3-7.

2. Alon N., Independent sets in regular graphs and Sum-Free Subsets of Finite Groups.// Israel Journal of Math., 73 (1991), No 2, 247-256.

3. Calkin N. On the number of sum-free set// Bull. London Math. Soc., 22 (1990), 141-144.

4. Cameron P., Erdos P., On the number of integers with various properties// in R. A. Mollin (ed). Number Theory: Proc. First Conf. Can. Number Th. Ass., Banff, 1988, — de Gruyter. 1990 — P. 61-79.

Системы языковых уравнений и замкнутые классы функций алгебры логики

А. С. Охотин (Кингстон, Канада)

1 Введение

Системы языковых уравнений, разрешённые относительно неизвестных и содержащие операции конкатенации и объединения, хорошо изучены [1] – в частности, известно, что они задают в точности класс контекстно-свободных языков. Рассматривая теоретико-множественное объединение как логическую связку – дизъюнкцию – и вводя дополнительные связки, можно получить классы систем с большей выразительной мощностью. Таковы системы над базисом из дизъюнкции и конъюнкции [6], эквивалентные конъюнктивным грамматикам [4, 5] – обобщению контекстно-свободных грамматик, допускающему использование явной операции пересечения в теле правил. Также изучались системы, содержащие связки из функционально полного множества из конъюнкции, дизъюнкции и отрицания [7], и системы, содержащие одно лишь отрицание [2].

В связи с этим естественно задаться вопросом описания систем языковых уравнений для *всех* возможных множеств логических связок. Опираясь на существующее описание всех замкнутых классов функций алгебры логики [3, 8, 9], задачу можно свести к рассмотрению 42 отдельных систем функций и 8 бесконечных иерархий таких систем. Этому рассмотрению посвящена данная работа.

2 Языковые уравнения

Теория замкнутых классов функций алгебры хорошо известна русскоязычному читателю по книгам [3, 9], и потому её терминология и основные положения повторяться не будут. Стараясь следовать её канонам, дадим определения языковых уравнений над данным базисом логических связок.

Определение 1 *Пусть $\mathcal{F} \subseteq P_2$ – множество (не обязательно конечное) функций алгебры логики. Пусть Σ – конечный непустой алфавит и пусть $X = (X_1, \dots X_n)$ $(n \geq 1)$ – вектор языковых переменных. Множество языковых формул над функциональным базисом $\mathcal{F}$ и над алфавитом Σ, зависящих от переменных X, определяется индуктивно следующим образом: (а) пустая строка ϵ – формула; (б) любой символ из Σ – формула; (в) любая переменная из X – формула; (г) если φ и ψ – формулы, то $\varphi\psi$ – формула;*

(д) для любой базисной функции k переменных $f^{(k)} \in \mathcal{F}$ и для любых формул $\varphi_1, \ldots, \varphi_k$, $f^{(k)}(\varphi_1, \ldots, \varphi_k)$ является формулой.

Как и в формулах алгебры логики, наиболее распространённые функции ($x\&y$, $x \vee y$, $x \oplus y$, $\neg x$) будут задаваться в операторной записи; при этом конкатенация имеет наибольший приоритет. Например, $\neg aX\&aY \vee XY$ означает то же, что $((\neg(a \cdot X))\&(a \cdot Y)) \vee (X \cdot Y)$, или $f_\vee(f_\&(f_\neg(aX), aY), XY)$.

Сопоставим теперь каждой формуле $\varphi(X_1, \ldots, X_n)$ $(n \geq 0)$ функцию f_φ, отображающую $(2^{\Sigma^*})^n$ в 2^{Σ^*}. Чтобы упростить запись, функция, соответствующая формуле φ, также обозначается φ, а её значения называются значениями формулы.

Определение 2 *Пусть φ – формула над базисом $\mathcal{F}$ и над алфавитом Σ, зависящая от переменных $X = (X_1, \ldots, X_n)$. Пусть $L = (L_1, \ldots, L_n)$ – вектор языков над алфавитом Σ. Значение φ на векторе L, обозначаемое $\varphi(L)$, определяется индуктивно по структуре φ: (а) $\epsilon(L) = \{\epsilon\}$. (б) $a(L) = \{a\}$ для любого $a \in \Sigma$. (в) $X_i(L) = L_i$ для любого i $(1 \leq i \leq n)$. (г) $\psi\xi(L) = \psi(L) \cdot \xi(L)$. (д) $f^{(k)}(\psi_1, \ldots, \psi_k)(L) = f^{(k)}(\psi_1(L), \ldots, \psi_k(L))$. Значение вектора формул $\varphi = (\varphi_1, \ldots, \varphi_l)$ на векторе языков $L = (L_1, \ldots, L_n)$ – вектор языков $\varphi(L) = (\varphi_1(L), \ldots, \varphi_l(L))$.*

Определение 3 *Пусть $\mathcal{F}$ – базис и пусть Σ – алфавит. Пусть $n \geq 1$. Пусть $X = (X_1, \ldots, X_n)$ – вектор языковых переменных. Пусть $\varphi = (\varphi_1, \ldots, \varphi_n)$ – вектор формул над базисом $\mathcal{F}$ и алфавитом Σ, зависящих от X. Тогда $X_i = \varphi_i(X_1, \ldots, X_n)$ $(1 \leq i \leq n)$ называется системой языковых уравнений над $\mathcal{F}$ и Σ, разрешённой относительно своих переменных X. Система также может задаваться в векторной форме: $X = \varphi(X)$.*

Определение 4 *Вектор языков $L = (L_1, \ldots, L_n)$ называется решением системы $X_i = \varphi_i(X_1, \ldots, X_n)$ $(1 \leq i \leq n)$, если для всякого i $(1 \leq i \leq n)$ выполняется равенство $L_i = \varphi_i(L_1, \ldots, L_n)$. В векторной форме это обозначается как $L = \varphi(L)$.*

L называется решением по модулю $M \subseteq \Sigma^*$, если $L_i \cap M = \varphi_i(L_1, \ldots, L_n) \cap M$.

Семантика систем языковых уравнений над базисом $\{x \vee y\}$, соответствующих контекстно-свободным грамматикам, задаётся с помощью минимального решения относительно покомпонентного включения [1]. Существование такого решения обусловлено монотонностью дизъюнкции, и потому эта семантика может быть расширена на случай любого монотонного базиса:

Теорема 1 *Пусть $\mathcal{F}$ – базис, состоящий из монотонных функций. Пусть $\varphi_i(X_1, \ldots, X_n)$ $(1 \leq i \leq n)$ – произвольные формулы. Тогда оператор $\varphi = (\varphi_1, \ldots, \varphi_n)$ на множестве $(2^{\Sigma^*})^n$ имеет наименьшую неподвижную точку вида $\sqcup_{i=0}^{\infty} \varphi^i(\emptyset, \ldots, \emptyset)$ и наибольшую неподвижную точку вида $\sqcap_{i=0}^{\infty} \varphi^i(\Sigma^*, \ldots, \Sigma^*)$.*

Следствие 1 *Всякая система языковых уравнений над базисом, состоящим из монотонных функций, имеет хотя бы одно решение, и среди её решений есть наибольшее и наименьшее. Это наименьшее решение можно использовать в качестве вектора языков, определяемого системой.*

Этот результат, однако, не может быть распространён ни на один немонотонный базис, и потому для систем над такими базисами необходимо использовать иную семантику. В

данной работе в качестве такой семантики используется *семантика единственного решения*, при которой система является корректной, если она имеет ровно одно решение, а первая компонента этого решения полагается языком, определяемым системой. Справедливы следующие результаты [7]:

Теорема 2 *Система $X = \varphi(X)$ над базисом $\{x \vee y, \neg x\}$ имеет решения тогда и только тогда, когда для любого конечного языка M, замкнутого относительно подстроки, система имеет решения по модулю M. Задача определения существования решений у данной системы со-***RE***-полна.*

Теорема 3 *Система $X = \varphi(X)$ над базисом $\{x \vee y, \neg x\}$ имеет единственное решение тогда и только тогда, когда для любого конечного языка M, замкнутого относительно подстроки, существует конечный язык $M' \supseteq M$, замкнутый относительно подстроки, такой что система имеет хотя бы одно решение по модулю M', и все решения по модулю M' равны по модулю M. Задача определения существования у данной системы единственного решения Π_2-полна.*

Следствие 2 *Если система $X = \varphi(X)$ над произвольным базисом $\mathcal{F}$ имеет единственное решение $L = (L_1, \ldots, L_n)$, то все L_i являются рекурсивными множествами.*

Класс языков, выразимых единственными решениями систем над базисом $\mathcal{F}$ и алфавитом Σ, будет обозначаться $\mathcal{F}^\Sigma$ ($\mathcal{F}^\Sigma \subseteq 2^{\Sigma^*}$).

Выразимость логических соотношений в системах языковых уравнений несколько отличается от выразимости функций формулами алгебры логики. Например, константу 0 можно выразить над базисом $\emptyset$, используя уравнение $Y = aY$ (единственное решение которого – пустое множество) и формулу $\psi = Y$. Константа 1 выразима над базисом из одной дизъюнкции, с помощью уравнения $X = a_1X \vee \ldots \vee a_nX \vee \epsilon$ (где $\Sigma = \{a_1, \ldots, a_n\}$), имеющего единственное решение Σ^*, означающее логическую единицу. Аналогичным образом константа 1 может быть выражена через сумму по модулю 2: $X = a_1X \oplus \ldots \oplus a_nX \oplus \epsilon$.

Определение 5 *Пусть Σ – некоторый алфавит. Пусть $\mathcal{F} \subseteq P_2$. Замыканием $\mathcal{F}$ относительно языковых уравнений над Σ называется множество $[\mathcal{F}]_\Sigma \subseteq P_2$ таких функций $f(x_1, \ldots, x_k)$ ($k \geq 0$), что существует система языковых уравнений $Y = \varphi(Y)$ над Σ и $\mathcal{F}$, имеющая единственное решение L, и языковая формула $\psi(X, Y)$ (где $X = (X_1, \ldots, X_k)$) над Σ и $\mathcal{F}$, такие что $\psi(X, L) \equiv f(X)$.*

Лемма 1 *Константы могут быть выражены как $0 \in [\emptyset]_\Sigma$, $1 \in [x \vee y]_\Sigma$, $1 \in [x \oplus y]_\Sigma$.*

Нетрудно видеть, что если $[\mathcal{F}_1] \subseteq [\mathcal{F}_2]$, то $[\mathcal{F}_1]_\Sigma \subseteq [\mathcal{F}_2]_\Sigma$. С другой стороны, из $[\mathcal{F}_1]_\Sigma \subseteq [\mathcal{F}_2]_\Sigma$ легко следует включение классов языков $\mathcal{F}_1^\Sigma \subseteq \mathcal{F}_2^\Sigma$, и потому количество таких классов не превосходит количества замкнутых классов функции алгебры логики.

3 Восемь классов языков

3.1 Класс U_0^Σ

Первый и самый узкий класс состоит из тех языков, которые могут быть выражены языковыми уравнениями вообще без использования логических связок. Как будет показано,

этот класс состоит из языков мощности не более 1 – т.е., из $\emptyset$ и из всех языков вида $\{w\}$, где $w \in \Sigma^*$. Кроме того, некоторые логических связки – например, конъюнкция – не увеличивают выразительную мощность этого класса.

Лемма 2 *Всякий язык из* $MI^{2,\Sigma}$ *имеет мощность не более 1.*

Доказательство (набросок). Пусть некоторая система языковых уравнений $X = \varphi(X)$ над базисом $\mathcal{F} = \{0,\ d_3(x,y,z)\}$ имеет единственное решение. Так как $MI^2 \subseteq M$, это решение является одновременно и наименьшим и потому может быть представлено в виде предела монотонно возрастающей последовательности векторов языков $\varphi^k(\overline{\emptyset}) = L^{(k)} = (L_1^{(k)}, \ldots, L_n^{(k)})$ $(k \geq 0)$.

Затем доказывается следующее *утверждение о единственности присваивания*: всякая i-я компонента либо принимает значение $\emptyset$ в каждом k-м элементе последовательности, либо равняется $\emptyset$ вплоть до некоторого $(p-1)$-го элемента $(p \geq 1)$, затем принимает значение некоторого языка мощности 1 на p-th шаге и сохраняет это значение во всех последующих элементах. Чтобы доказать невозможность переприсваивания, достаточно рассмотреть *наименьшее* p, такое что на p-м шаге переприсваивается некоторая i-я компонента, и, рассмотрев уравнение для X_i, вывести отсюда, что какие-то переменные должны были изменить непустое значение на $(p-1)$-м шаге, что противоречит предположению о минимальности p.

Отсюда следует, что предел последовательности – и, стало быть, решение системы – состоит из языков мощности не более 1.

Теорема 4 *Следующие классы совпадают:*

$$MI^{k,\Sigma},\ MI_1^{k,\Sigma}\ \ (\textit{для всех } k \geq 2)\ ,\ MI^{\infty,\Sigma},\ MI_1^{\infty,\Sigma},\ K_0^\Sigma,\ K_{01}^\Sigma,\ SM^\Sigma, U_0^\Sigma,\ U_{01}^\Sigma,\ C_0^\Sigma\ \ \textit{и}\ \emptyset^\Sigma.$$

3.2 Класс U_1^Σ

Этот класс состоит из языков вида

$$\emptyset \quad \text{или} \tag{1}$$

$$w_0\Sigma^* w_1\Sigma^* \ldots w_{n-1}\Sigma^* w_n \quad (n \geq 0,\ w_i \in \Sigma^*) \tag{2}$$

Лемма 3 *Всякий язык из* MU^Σ *имеет вид* (1).

Доказательство (набросок). Любой язык вида (2) может быть задан с использованием логической константы 1. Доказательство проводится по той же схеме, что и доказательство леммы 2: утверждается, что в монотонной последовательности $\varphi^k(\overline{\emptyset}) = L^{(k)} = (L_1^{(k)}, \ldots, L_n^{(k)})$ каждая i-я компонента или равна $\emptyset$ во всех элементах, или равна $\emptyset$ в конечном числе элементов, а затем принимает значение вида (2), которое сохраняет во всех последующих элементах.

Теорема 5 $MU^\Sigma = U_1^\Sigma = C^\Sigma = C_0^\Sigma$.

3.3 Класс K^Σ

Этот класс систем получается добавлением конъюнкции в U_1^Σ.

Теорема 6 *$K^\Sigma = K_1^\Sigma$. Для любого алфавита Σ, класс K^Σ совпадает с замыканием класса $U_0^\Sigma \cup \{\Sigma^*\}$ относительно конкатенации и пересечения.*

Теорема 6 доказывается тем же методом, что лемма 3.

3.4 Класс U^Σ

Класс U^Σ состоит из языков, определяемых языковыми уравнениями, использующими *только* отрицание. Для случая унарного алфавита такие системы были рассмотрены в [2], где показывалось, что с их помощью может быть задан некоторый нерегулярный и потому не-контекстно-свободный язык – язык $L = \{a^n \mid 2^{3k} \leq n < 2^{3k+2}$ для некоторого $k \geq 0\}$, являющийся единственным решением уравнения $X = a(\neg(\neg(\neg X)^2)^2)^2$ (где φ^2 – это сокращённая запись $\varphi \cdot \varphi$).

Теорема 7 *$U^\Sigma = SU^\Sigma$.*

Доказательство. Так как $SU = [\neg x]$ и $U = [0, \neg x]$, достаточно выразить константу 0 в системах над SU с помощью леммы 1.

3.5 Класс L^Σ

Этот класс состоит из языков, задаваемых с использованием одной лишь суммы по модулю два. Такие системы внешне довольно похожи на классические системы языковых уравнений [1], в которых сумма по модулю 2 заменена дизъюнкцией.

Лемма 4 *Функции $x \oplus y$ и 1 выразимы в системах над базисом L_{01}.*

Доказательство. Рассмотрим конечный базис $\{f(x, y, z) = x \oplus y \oplus z\}$. Константа 0 выразима из ничего. Сумма по модулю 2 двух переменных представима в виде $g(x, y) = f(x, y, 0) = x \oplus y$. Затем константа 1 может быть выражена по методу леммы 1.

Теорема 8 *$L^\Sigma = SL^\Sigma = L_0^\Sigma = L_1^\Sigma = L_{01}^\Sigma$. Всякий язык, порождаемый однозначной контекстно-свободной грамматикой в н.ф. Грейбах [1], принадлежит L^Σ.*

Доказательство (набросок). Рассмотрим однозначную контекстно-свободную грамматику, в которой каждое правило имеет вид $A \to a\alpha$ (где $a \in \Sigma$ и $\alpha \in (\Sigma \cup N)^*$) или $A \to \epsilon$. Система $X_i = \alpha_{i1} \vee \ldots \vee \alpha_{ik_i}$ $(1 \leq i \leq n)$, соответствующая этой грамматике – это строгая система [1], имеющая единственное решение. Заменяя в ней дизъюнкцию суммой по модулю 2, получаем систему $X_i = \alpha_{i1} \oplus \ldots \oplus \alpha_{ik_i}$ $(1 \leq i \leq n)$, единственность решения которой может быть доказана тем же методом [1]. Используя однозначность исходной грамматики, можно доказать, что эти решения совпадают.

Заметим, что $U^\Sigma \subseteq L^\Sigma$, и потому L^Σ содержит не-контекстно-свободные языки.

3.6 Класс D^Σ

Это классические языковые уравнения, в которых используется дизъюнкция и, возможно, логические константы [1]. Так как дизъюнкция позволяет выразить единицу и ноль, наличие логических констант в базисе не увеличивает выразительной мощности.

Теорема 9 *$D^\Sigma = D_0^\Sigma = D_1^\Sigma = D_{01}^\Sigma$. D^Σ совпадает с классом контекстно-свободных языков.*

3.7 Класс M^Σ

Системы языковых уравнений над базисом $M_{01} = \{x \vee y, x\&y\}$ были изучены в [6], где было показано, что их выразительная мощность относительно семантики наименьшего решения совпадает с выразительной мощностью конъюнктивных грамматик [5]. Распространим этот результат на некоторые другие базисы.

Лемма 5 *Конъюнкция, дизъюнкция, единица и ноль выразимы в $MO_0^{\infty,\Sigma}$.*

Доказательство. Рассмотрим базис $\{x \vee y\&z\}$ в классе MO_0^∞. Отождествим y и z, чтобы получить дизъюнкцию $f(x,y) = x \vee y\&y = x \vee y$, а затем получим из дизъюнкции единицу, используя лемму 1. Согласно той же лемме 1, ноль может быть выражен без использования связок. Затем конъюнкция выражается из $x \vee y\&z$, положив $x = 0$.

Теорема 10 *Следующие классы совпадают: M^Σ, M_0^Σ, M_1^Σ, M_{01}^Σ, $MO^{k,\Sigma}$ и $MO_0^{k,\Sigma}$ (для всех $k \geq 2$), $MO^{\infty,\Sigma}$ и $MO_0^{\infty,\Sigma}$. M^Σ совпадает с классом языков, порождаемых конъюнктивными грамматиками.*

3.8 Класс P_2^Σ

Лемма 6 *Конъюнкция и отрицание выразимы в системах над базисом S_{01}.*

Доказательство. Пусть $f(x,y,z) = d_3(\neg x, y, z) = \neg x\&y \vee y\&z \vee \neg x\&z$ – единственная базисная функция. Подставим f саму в себя следующим образом:

$$g(x,y,z) = f(x, f(y,x,z), f(z,x,y)) = x \oplus y \oplus z.$$

Теперь, используя лемму 4, мы можем выразить $x \oplus y$ и 1. Это позволяет получить конъюнкцию как $h(y,z) = f(1,y,z) = y\&z$ и отрицание в виде $x \oplus 1$.

Лемма 7 *Дизъюнкция и отрицание выразимы в системах над базисом O_0^∞.*

Доказательство. Рассмотрим базисную функцию $f(x,y,z) = x \vee y\&\neg z$. Положив $x = z$, получаем $g(x,y) = x \vee y\&\neg x = x \vee y$. По лемме 1, из дизъюнкции выражается единица, а ноль может быть получен просто так. Остаётся выразить отрицание подстановкой в f констант: $f(0,1,z) = h(z) = \neg z$.

Лемма 8 *Конъюнкция и отрицание выразимы в системах над базисом I_1^∞.*

Доказательство. Пусть базис состоит из единственной функции $f(x,y,z) = x\&(y \vee \neg z)$. Ноль выражается по лемме 1 и может затем быть подставлен в f в качестве второго аргумента, что даёт отрицание импликации $g(x,z) = f(x,0,z) = x\&\neg z$.

Теперь, используя функцию $x\&\neg z$, получим константу 1. Пусть $\Sigma = \{a_1, \ldots, a_n\}$ – алфавит; построим следующую систему из $2n+3$ языковых уравнений: $A = A$, $B_i = a_i A\&\neg A$ $(1 \leq i \leq n)$, $T_i = \neg T_i \& B_i$ $(1 \leq i \leq n)$, $C = \neg A\&\epsilon$, $D = \neg D\&C$. Нетрудно доказать, что единственное решение этой системы – вектор $(\Sigma^*, \emptyset, \emptyset, \ldots, \emptyset, \emptyset, \emptyset, \emptyset)$, и потому переменная A определяет искомую единицу. Затем отрицание может быть выражено как $f(1,0,z) = \neg z$, а конъюнкция – как $f(x,y,1) = x\&y$.

Теорема 11 *Следующие классы совпадают:* P_2^Σ, T_0^Σ, T_1^Σ, T_{01}^Σ, S^Σ, S_{01}^Σ, $I^{k,\Sigma}$, $I_1^{k,\Sigma}$, $O^{k,\Sigma}$ *and* $O_0^{k,\Sigma}$ *(для всех* $k \geq 2$*),* $I^{\infty,\Sigma}$, $I_1^{\infty,\Sigma}$, $O^{\infty,\Sigma}$ *and* $O_0^{\infty,\Sigma}$*. Для всякого алфавита* Σ*, такого что* $|\Sigma| \geq 2$*,* P_2^Σ *совпадает с классом рекурсивных языков над* Σ*.*

Доказательство. Пусть M – произвольная машина Тьюринга, получающая на входе строки над алфавитом Σ и останавливающаяся на любом входе. Построим систему языковых уравнений над базисом $\mathcal{F} = \{x \vee y, x\&y, \neg x\}$, такую что первая компонента её единственного решения будет совпадать с $L(M)$.

Пусть Q – множество состояний машины M, а $V \supset \Sigma$ – её рабочий алфавит. Рассмотрим алфавит $\Gamma = V \cup Q \cup \{\#, /\}$. Пусть k=$\lceil \log_2 |\Gamma| \rceil$ и пусть 0 и 1 – два различных символа из Σ. Определим двоичную кодировку символов из Γ – гомоморфизм $h : \Gamma^* \to \{0,1\}^*$, такой что $|h(a)| = k$ для любого символа $a \in \Gamma$ и $h(a) \neq h(b)$ для всех $a \neq b$ $(a, b \in \Gamma)$. Пусть $ID_M(w,i)$ – конфигурация M на i-м шаге вычисления на строке w, заданная в виде строки $\alpha a q \beta$ $(\alpha, \beta \in V^*, a \in V, q \in Q)$ над алфавитом $V \cup Q$. Рассмотрим следующие два языка:

$$L_{Acc.Comp.} = \{wh(/)h(ID_M(w,0))h(\#)\ldots h(\#)h(ID_M(w,k)) \mid ID_M(w,k) \text{ – принимающая конфигурация}\} \quad (5)$$

$$L_{Rej.Comp.} = \{wh(/)h(ID_M(w,0))h(\#)\ldots h(\#)h(ID_M(w,k)) \mid ID_M(w,k) \text{ – отвергающая конфигурация}\} \quad (6)$$

Используя метод [5], легко построить конъюнктивные грамматики для обоих языков, откуда, согласно [6], следует существование систем языковых уравнений $X = \varphi(X)$ и $Y = \psi(Y)$ над базисом $\{x \vee y, x\&y\}$ и над алфавитом Σ, каждая из которых имеет единственное решение (L' и L'' соответственно), и $L'_1 = L_{Acc.Comp.}$, $L''_1 = L_{Rej.Comp.}$.

Построим систему, состоящую из следующих уравнений: $Z = Z, R = \epsilon \vee \bigvee_{a \in \Gamma \setminus \{/\}} h(a)R$, $T_1 = \neg T_1 \& X_1 \& \neg\, Z\, h(/)R$, $T_2 = \neg T_2 \& Y_1 \& \neg\, (\neg Z) h(/) R$, а также из уравнений, входящих в $X = \varphi(X)$ и $Y = \psi(Y)$. Нетрудно видеть, что $(L(M), (\bigcup_{a \neq /} h(a))^*, \emptyset, \emptyset, L', L'')$ – её единственное решение. Действительно, если $w \in L(M)$, то в $L_{Acc.Comp.}$ есть строка, начинающаяся с w, и если при этом переменная Z не содержит w, то уравнение для T_1 становится противоречием. Аналогично, если $w \notin L(M)$, то язык $L_{Rej.Comp.}$ содержит строку, начинающуюся с w, и потому Z не может содержать w в соответствии с уравнением для T_2.

4 Заключение

Было установлено, что языковые уравнения над различными базисами логических связок определяют восемь классов языков (см. рис. 1): это $U_0^\Sigma \subset U_1^\Sigma \subset K^\Sigma \subset D^\Sigma \subset M^\Sigma \subset P_2^\Sigma$, а

также U^{Σ} и L^{Σ}. Относительно последних известно лишь, что $U_1^{\Sigma} \subset U^{\Sigma} \subseteq L^{\Sigma} \subseteq P_2^{\Sigma}$, а также $K^{\Sigma} \subset L^{\Sigma}$. Подробное изучение свойств этих классов предлагается в качестве открытой проблемы.

Литература

[1] J. Autebert, J. Berstel, L. Boasson, "Context-Free Languages and Pushdown Automata", *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, Berlin, 1997, 111–174.

[2] E. L. Leiss, *Language equations*, Springer-Verlag, New York, 1999.

[3] С. С. Марченков, *Замкнутые классы булевых функций*, Физматлит, Москва, 2000.

[4] А. Охотин, "О расширении формализма контекстно-свободных грамматик операцией пересечения", *Труды IV Международной конференции "Дискретные модели в теории управляющих систем"*, 2000, 106–109.

[5] A. Okhotin, "Conjunctive grammars", *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[6] А. Охотин, "Конъюнктивные грамматики и системы языковых уравнений", *Программирование*, 28:5, 2002.

[7] A. Okhotin, "Decision problems for language equations with Boolean operations", *Proceedings of ICALP 2003*, LNCS, to appear.

[8] E. L. Post, *The two-valued iterative systems of mathematical logic*, Princeton University Press, 1941.

[9] С. В. Яблонский, Г. П. Гаврилов, В. Б. Кудрявцев, *Функции алгебры логики и классы Поста*, Наука, Москва, 1966.

Двухступенчатое моделирование программ с процедурами

Р. И. Подловченко, Б. А. Долгих (Москва)

Доклад относится к теории моделей программ – одному из разделов теоретического программирования (см.[1]).

Исходными в этой теории являются выбор той или иной формализации программы и задача построения эквивалентных преобразований (э.п.) программ. Решение этой основной задачи ищется на следующем пути:

- запись программы транслируется к виду, когда ее управляющая структура представлена ориентированным графом;
- осуществляется переход от программ к схемам, сохраняющим управляющую структуру программ;
- в множестве схем программ вводится отношение эквивалентности схем; выбор его квалифицируют как рассмотрение конкретной модели программ;
- среди различных моделей выделяются аппроксимирующие исходный класс программ; в такой модели из эквивалентности схем следует функциональная эквивалентность программ, представленных этими схемами;

- для аппроксимирующих моделей в качестве главной рассматривается проблема построения системы э.п. схем, полной в модели.

Отметим, что всякое э.п. схемы, принадлежащей аппроксимирующей модели, одновременно является и э.п. программы, представленной этой схемой. Отсюда: система э.п. схем, полная в модели, является наилучшим из решений основной задачи для программ, если в поисках решений мы ограничиваемся рамками данной модели.

Здесь нами рассматривается следующая формализация программы: последняя записана на алгоритмическом языке высокого уровня в условиях, когда фиксируется базис операторов присваивания и булевых выражений, используемых программой, и допускаются все традиционные композиции операторов, включая процедуры. Прежде всего осуществляется запись программы на графе её управляющей структуры. Далее для таких программ выполнено двухступенчатое построение аппроксимирующих моделей, а именно: сначала построены схемы программ, получившие название стандартных, затем – схемы программ, названные алгебраическими.

Стандартная схема получается из программы заменой базисных операторов присваивания и булевых выражений конструкциями, в которых полностью сохраняется информация об использованных переменных, а конкретные операции и отношения уступают место произвольно трактуемым; такие конструкции называются операторами и предикатами над памятью. Их структура однозначно определяет отношение эквивалентности стандартных схем. Модель, образованная стандартными схемами, является аппроксимирующей.

Алгебраическая схема строится по стандартной путём замены операторов над памятью операторными символами, а предикатов над памятью – логическими переменными; кроме того, опускаются определённые характеристики стандартной схемы.

Заметим теперь, что, в отличие от стандартных схем с однозначно вводимым отношением эквивалентности, для алгебраических схем отношения эквивалентности составляют параметрическое множество.

Нами была решена следующая задача: определены параметры, которые индуцируют эквивалентность алгебраических схем, удовлетворяющую требованию: две алгебраические схемы, построенные по стандартным, эквивалентны тогда и только тогда, когда эквивалентны эти стандартные схемы. Данным результатом мы получаем множество аппроксимирующих моделей, ибо для эквивалентностей алгебраических схем имеется удобный достаточный признак того, когда одна эквивалентность влечёт другую.

В заключение отметим, что идея двухступенчатого моделирования программ почерпнута из [2], где рассматривается формализм программы, отличающийся от нашего запретом на использование процедур. Именно привлечение процедур потребовало решения ряда специфических задач для реализации этой идеи.

Литература

1. Подловченко Р.И. От схем Янова к теории моделей программ // Математические вопросы кибернетики, Физматгиз, вып.7, 1998, с. 281-302.

2. Подловченко Р.И. Эквивалентные преобразования схем программ для “запутывания” самих программ // Программирование, 2002, N2, с. 66-80.

Каноническая форма схемы программ с однократным вхождением константы

Р. И. Подловченко, Д. М. Русаков (Москва)

Доклад относится к теории моделей программ, являющейся одним из разделов теоретического программирования (см. [1]). Объектами этой теории служат схемы программ, формализованные тем или иным путем. Введение в множестве схем отношения их эквивалентности приводит к конкретной модели программ. Проблематика теории складывается вокруг задачи построения систем эквивалентных преобразований схем программ. Обычно системы конструируются для классов схем программ с разрешимой проблемой эквивалентности. Отсюда - повышенный интерес к изучению этой проблемы. Поиски ее разрешения наталкиваются на выделение канонических форм схем программ. Такие формы составляют множество, обладающее свойством: в каждом классе эквивалентности схем имеется каноническая форма, и она единственна с точностью до изоморфизма.

Опишем рассматриваемые нами модель программ и принадлежащий ей класс схем, а также задачи, решаемые для этого класса.

Из различных формализаций схемы программы мы выбрали формализацию, которая дает схемы, называемые алгебраическими. Такая схема строится над алфавитом Y операторных символов и алфавитом P логических переменных; каждая из последних принимает значения 0 и 1. Структурно схема представляет собой конечный ориентированный граф с выделенным входом - вершиной без входящих дуг и с одной исходящей и с выделенным выходом - вершиной без исходящих дуг; остальные вершины графа, если таковые имеются, подразделяются на преобразователи и распознаватели. Из преобразователя исходит одна дуга, и ему сопоставлен символ из Y. Из распознавателя исходят две дуги, помеченные 0 и 1 соответственно, и ему сопоставлена переменная из P.

Функционирование схемы осуществляется на функциях, которые цепочке из $Y^\star$ сопоставляют значения всех переменных из P; такая функция называется функцией разметки.

Выполнение схемы на функции разметки представляет собой путешествие по схеме, сопровождающееся накоплением цепочки из $Y^\star$. Оно начинается во входе схемы с пустой операторной цепочкой и совершается по правилам: при переходе через преобразователь с символом y к текущей цепочке приписывается справа y; переход через распознаватель не меняет текущей цепочки, если p - приписанный ему символ, то выход из распознавателя осуществляется по дуге, которая помечена значением, сопоставленным переменной p функцией разметки на текущей цепочке. Если достигнут выход схемы, то говорим, что схема остановилась на данной функции разметки, и текущую цепочку называем результатом выполнения схемы на данной функции разметки.

Чтобы определить эквивалентность схем , выбираются: отношение эквивалентности в $Y^\star$ и множество функций разметки, называемых допустимыми. Пусть эти параметры выбраны; тогда две схемы считаются эквивалентными, если, какой бы ни была допустимая функция разметки, всякий раз, как на ней останавливается одна из схем, останавливается и другая, и результат их выполнения - это цепочки, эквивалентные по выбранному отношению.

В нашем случае эквивалентность в $Y^\star$ определяется так: в Y выделяется один символ, называемый константой; обозначим его c; две цепочки считаем эквивалентными, если выполнены условия: либо обе цепочки не имеют вхождения символа c, либо обе имеют;

в первом случае совпадают сами цепочки, а во втором – их постфиксы, начинающиеся последним вхождением символа c. Допустимой, по определению, является функция разметки, удовлетворяющая требованию: эквивалентным цепочкам она сопоставляет равные значения p, и так для всех p из P.

В модели программ, индуцированной выбранными параметрами, рассматривается класс схем, каждая из которых имеет не более одного преобразователя, помеченного символом c. Он называется классом схем программ с однократным вхождением константы. Обозначим его M.

Ставятся проблемы:

- показать, что для класса M могут быть определены канонические формы схем;
- построить алгоритм, транслирующий произвольную схему из M в эквивалентную ей каноническую форму.

Обе проблемы решены.

Доказано, что канонической является унифицированная схема (см. [1]), удовлетворяющая требованиям:

1. пути из входа схемы в вершину с константой, и пути, ведущие из нее в выход схемы, не имеют других общих вершин, кроме самой вершины с константой;
2. отсутствуют преобразователи, x-преемниками которых при всех x из X, является вершина с константой;
3. любые два различных преобразователя не эквивалентны.

Здесь $X = \{x | x : P \to \{0, 1\}\}$, а эквивалентность вершин схемы определяется подобно тому, как эквивалентность состояний конечных автоматов.

Литература

1. Подловченко Р.И. От схем Янова к теории моделей программ // Математические вопросы кибернетики, Физматлит, вып. 7, 1998, с. 281-302.

Алгоритм распознавания эквивалентности многоленточных автоматов без пересекающихся циклов

Р. И. Подловченко (Москва), В. Е. Хачатрян (Белгород)

Нами рассматриваются многоленточные автоматы [1],представленные диаграммами. Такое их представление восходит к работе [2], впервые давшей алгоритм распознавания эквивалентности двухленточных автоматов. Существование распознающего эквивалентность алгоритма для автоматов с большим числом лент доказано в работе [3]. Возникла задача построения самого алгоритма, основанного на изучении структуры эквивалентных автоматов. Мы даем решение этой задачи для класса диаграмм, не имеющих пересекающихся циклов. Заметим, что в работе [4] для него построена полная система эквивалентных преобразований.

Напомним, что диаграмма строится над алфавитами P и Q, где

$$P = \{p_1, ..., p_n\}, n \geq 1; Q = \{0, 1\},$$

и представляет собой конечный ориентированный граф, из каждой вершины которого исходят по две дуги, помеченные 0 и 1 соответственно, за исключением одной вершины, называемой финальной: из нее нет исходящих дуг. Среди вершин, отличных от финальной, одна выделена как инициальная. Все вершины, кроме финальной, помечены символами из P . Всякий маршрут из инициальной вершины в финальную описывается историей вида

$$(p_{i_1}, \varepsilon_{i_1})(p_{i_2}, \varepsilon_{i_2})...(p_{i_k}, \varepsilon_{i_k}), \tag{1}$$

где p_{i_j} - метка вершины, из которой исходит j-ая дуга маршрута, а ε_{i_j} - метка самой дуги; здесь $j = 1, ..., k$, где k - общее число дуг в маршруте. Две диаграммы над P , Q , по определению, эквивалентны, если для всякого маршрута через одну из них в другой имеется маршрут, обладающий свойством: для любого p из P совпадают проекции первого и второго маршрута на символ p. Проекция (1) на p получается вычеркиванием из (1) всех пар с первой компонентой, отличной от p.

Нами рассматриваются диаграммы без пересекающихся циклов. Действия предлагаемого алгоритма распознавания эквивалентности таких диаграмм основаны на ряде фактов. Описывая этот алгоритм, мы будем их формулировать без доказательства.

Итак, пусть D_1, D_2 - диаграммы из взятого нами класса, поступившие на вход алгоритма. Рангом диаграммы будем называть максимальное число циклов, посещаемых маршрутами через диаграмму.

Утверждение 1. *Если D_1, D_2 эквивалентны, то их ранги совпадают. Алгоритм проверяет ранги диаграмм D_1, D_2 и останавливается с ответом "нет", если они не равны. В противном случае он продолжает свою работу. Обозначим n общий ранг D_1, D_2.*

К D_1, D_2 применяется процедура ρ, нацеленная на построение из диаграммы D_2 диаграммы D того же ранга; при этом ρ руководствуется информацией о D_1.

Утверждение 2. *Если процедура ρ отказывается от построения D, то D_1, D_2 не эквивалентны.*

В этой ситуации алгоритм останавливается с ответом "нет". Предположим, что процедура ρ завершилась построением D.

Утверждение 3. *Если D_1, D_2 эквивалентны, и $n = 0$, то диаграммы D_1, D изоморфны.*

Отсюда: если D_1, D изоморфны, то алгоритм останавливается с ответом "да", в противном случае - с ответом "нет".

Пусть $n > 1$. Тогда алгоритм строит непустое множество

$$(d_1, d_1'), (d_2, d_2'), ..., (d_k, d_k'), \tag{2}$$

состоящее из пар вершин диаграммы D.

Утверждение 4. *D_1, D_2 эквивалентны тогда и только тогда, когда каждая пара из* (2) *состоит из эквивалентных вершин.*

Отметим, что, по определению, вершины d_1, d_2 диаграммы D эквивалентны, если эквивалентны диаграммы $D(d_1)$, $D(d_2)$, где последние - это поддиаграммы диаграммы D, в коих инициальными являются d_1, d_2 соответственно.

Всякая пара (d, d') из (2) подвергается алгоритмом проверке: изоморфны или нет диаграммы $D(d), D(d')$. В первом случае d, d' эквивалентны, во втором (d, d') аттестуется как головная пара. Если в (2) отсутствуют головные пары, то алгоритм останавливается с ответом "да". Иначе он рассматривает каждую головную пару в отдельности.

Пусть (d, d') - головная пара. Тогда к ней применяется процедура τ, которая при успешном ее завершении строит диаграмму C.

Утверждение 5. *Если процедура τ отказывается от построения C, то вершины d, d' не эквивалентны.*

В случае отказа τ алгоритм останавливается с ответом "нет". В противном случае он строит непустое множество

$$(c_1, c_1'), (c_2, c_2'), ..., (c_l, c_l') \tag{3}$$

пар вершин из диаграммы C.

Утверждение 6. *Вершины d, d' эквивалентны тогда и только тогда, когда каждая пара из* (3) *состоит из эквивалентных вершин.*

Условимся пары из (3) называть потомками пары (d, d').

Каждый потомок проверяется алгоритмом на выполнимость для него свойства α, которое выявляется процедурой линейной сложности.

Утверждение 7. *Если потомок (c, c') обладает свойством α, то c, c' эквивалентны.*

Если все потомки обладают свойством α, то алгоритм переходит к рассмотрению другой головной пары, если таковые имеются. Если они исчерпаны, то он останавливается с ответом "да".

Пусть (c, c') не обладает свойством α. Тогда к этой паре применяется процедура τ. Если она не увенчалась успехом, то c, c' не эквивалентны, и алгоритм останавливается с ответом "нет". В случае, когда τ завершилась построением диаграммы, определяются пары принадлежащих ей вершин, которые являются потомками пары (c, c'). Для них и самой (c, c') справедливо утверждение, аналогичное утверждению 6. Потомки (c, c') именуются потомками второго поколения пары (d, d'). С ними алгоритм проводит ту же работу, что и с потомками первого поколения.

Справедлива

Теорема 1. *Если d, d' не эквивалентны, то встретится потомок, поколение которого не превышает числа $n + 2$, и для которого процедура τ неуспешна. В случае, когда d, d' эквивалентны, алгоритм построит для пары (d, d') дерево потомков высоты, не превышающей $n + 2$, и все листья которого обладают свойством α.*

В первом случае алгоритм остановится с ответом "нет", а во втором перейдет к работе с новой головной парой. При исчерпании их алгоритм дает ответ "да".

Предлагаемый нами алгоритм полностью описан. При его построении использован анализ эквивалентных диаграмм, выполненный в [4] и [5].

Литература

1. Rain M.O., Scott D. Finite automata and their decision problems // IBM Journal of Research and Development. 1959. v.3. N 2. P. 114-125.

2. Bird R. The equivalence problem for deterministic two-tape automata // Journal of Computer end System Science. 1973. v.7. N 4. P. 218-236.

3. Harju T.,Karhumaki J.The equivalence of multitape finite automata // Theoretical Computer Science. 1991. v.78. N 2. P.347-355.

4. Подловченко Р.И., Хачатрян В.Е., Чашин Ю.Г. Полная система эквивалентных преобразований для двухленточных автоматов с непересекающимися циклами // Программирование. 2000. N 5. С. 3-17.

5. Хачатрян В.Е. Полная система эквивалентных преобразований для многоленточных автоматов // Программирование. 2003. N 1. С. 66-78.

Сложность оптимального по точности алгоритма вычисления сингулярного интеграла

Т. И. Полякова (Пенза)

Рассмотрим сингулярный интеграл:

$$Kw = \frac{1}{2\pi}\int\limits_0^{2\pi} w(\sigma)\frac{1 - rcos(\sigma - s)}{1 + r^2 - 2rcos(\sigma - s)}d\sigma. \qquad (1)$$

Интеграл (1) имеет широкое применение в различных областях физики и техники. Так как точное вычисление этого интеграла возможно лишь в исключительных случаях, то возникает необходимость в разработке приближенных методов вычисления.

В работе оценена сложность асимптотически оптимальной по точности квадратурной формулы вычисления интеграла, построенной в работе [2] на классе функций Гельдера $H_\alpha(1)$, $0 < \alpha \leq 1$.

Возьмем в качестве набора простейших операций набор
p={арифметические операции, вычисление значения функции [1].

Пусть $\eta_N = (w(t_1), \ldots, w(t_N))$ - информационный оператор, допустимый по отношению к P. Через $comp(\eta_N(w))$ обозначается информационная сложность вычисления $\eta_N(w)$.

Сложность квадратурной формулы

$$Kw = \sum_{k=1}^{N} w(t_k)p(r,s) + R_N(r,s,t_k,p_k(r,s),w) \qquad (2)$$

определяется равенством

$$comp(k) = \sup_{w \in H_\alpha(1)} (comp(\eta_N(w))) + comp(K(\eta_N(w))).$$

Определение. [1] Величина $comp(\eta_N, K, \epsilon)$, задаваемая формулой

$$comp(K) = \left\{ \begin{array}{c} \inf comp(K),\ если \inf\limits_{(t_k,p_k)} \sup\limits_{w \in H_\alpha(1)} \max\limits_{(r,s)} |R_N| < \epsilon\ u\ A(\epsilon) \neq 0 \\ +\infty, \end{array} \right\}$$

в противном случае называется ϵ - сложностью задачи K при использовании информации η_N. Здесь $A(\epsilon)$ представляет собой класс всех допустимых квадратурных формул, для которых погрешность меньше ϵ.

Теорема. [2] *Пусть* $\Lambda = H_\alpha(1)$, $0 \le r \le \rho \le 1 - \frac{ln^v N_0}{N_0}$, $0 < v < 1$, N_0 - *целое число. Тогда среди всевозможных квадратурных формул вида (2), асимптотически оптимальной является формула*

$$Kw = \frac{1}{2\pi} \sum_{k=0}^{N-1}{}' w(t'_k) \int_{t_k}^{t_{k+1}} \frac{1 - rcos(\sigma - s)}{1 + r^2 - 2rcos(\sigma - s)} d\sigma +$$

$$+ \frac{1}{2\pi} \int_{t_{j-1}}^{t_{j+2}} w(t'_j) \frac{1 - rcos(\sigma - s)}{1 + r^2 - 2rcos(\sigma - s)} + R_N, \qquad (3)$$

где $t'_k = \frac{(2k+1)\pi}{N}$, $t_k = \frac{2k\pi}{N}$.

Погрешность R_N формулы (3) равна

$$|R_N[H_\alpha(1)]| = \frac{\pi^\alpha}{(1+\alpha)N^\alpha} + o(N^{-\alpha}).$$

Оценим вначале число простейших операций, необходимых для реализации квадратурной формулы (2) при произвольной $s(0 \le s \le 2\pi)$, фиксированном r и без предварительной обработки коэффициентов. Выше было показано, что для информационного оператора

$$\eta_N = (w(t_1), \ldots, w(t_N)) \quad \xi_N[H_\alpha(1)] \ge (1 + o(1)) \frac{\pi^\alpha}{(1+\alpha)N^\alpha}.$$

Из определения функционала $\xi_N[H_\alpha(1)]$ следует, что нижняя грань размерности оператора η_N, необходимая для достижения точности ϵ, определяется из неравенства $\xi_N[H_\alpha(1)] < \epsilon$. Отсюда следует, что размерность оператора η_N должна быть не меньше $N = A(1/\epsilon)^{1/\alpha}$, а при $N \to \infty$

$$N = (1 + o(1)) \left(\frac{\pi^\alpha}{(1+\alpha)\epsilon} \right)^{1/\alpha}.$$

Оценка снизу сложности к.ф. (2) $comp(\eta_N, K, \epsilon)$ равна $\sum_{i=1}^{N} (c_i + a_i) + N_1$, где c_i - сложность вычисления каждого функционала $w(t_i)$, $1 \le i \le N_0$, a_i - сложность вычисления каждой весовой функции $p_i(r, s)$. Очевидно, $c_i \ge 2$, $a_i \ge 0$, $N_1 \ge N$. Следовательно, $comp(\eta_N, K, \epsilon) \ge 2(1 + o(1))N$. Число арифметических действий, необходимое для реализации к.ф. (3), не превосходит $3(1 + o(1))N$. При оценке сложности был применен метод, предложенный в [3].

Литература

1. Трауб Дж., Вожьняковский Х. Общая теория оптимальных алгоритмов. - М.: Мир, 1983. 382 с.

2. Бойков И.В., Полякова Т.И. Асимптотически оптимальные алгоритмы вычисления интегралов Пуассона, Шварца и типа Коши // Оптимальные методы вычислений и их применение: Межвуз. сб. науч.тр. - Пенза: Изд-во Пенз. гос. техн. ун-та, 1996. - Вып. 12 - с. 20-44.

3. Бойков И.В. Пассивные и адаптивные алгоритмы приближенного вычисления сингулярных интегралов. Части 1,2. Пенза: Изд-во Пенз. гос. техн. ун-та, 1995. - 342 с.

Об одном обобщении пропозиционального языка

С. В. Попов (Москва)

Рассматривается расширение обычного пропозиционального языка за счет введения многоместных обобщений дизъюнкции и конъюнкции на не более, чем счетное число аргументов. Для многоместных дизъюнкции и конъюнкции используются выражения соответственно $\cap_{i=k,h}F_i$ и $\cup_{i=k,h}F_i$. При этом k и h конечны или бесконечны, k называется *нижней* границей, а h - *верхней*. Верхняя граница всегда не меньше нижней и может иметь бесконечное значение, нижняя граница - всегда конечное неотрицательное число.

Множество всех переменных произвольной формулы F обозначим $Var(F)$. Логические значения функций в расширенном базисе вычисляются путем расширения соответствующих таблиц истинности: обобщенная дизъюнкция истинна тогда и только тогда, когда истинен, по меньшей мере, один ее аргумент, а обобщенная конъюнкция тогда и только тогда, когда истинны все ее аргументы.

Две формулы логически эквивалентны, если множества означиваний их переменных, при которых эти формулы истинные, совпадают.

Ограничим синтаксис обобщенных формул за счет следующих условий.

Для подформул $\cap_{i=k,h}F_i$ и $\cup_{i=k,h}F_i$ индекс i в каждой формуле F_i относится только к переменным или определяет границы обобщенных связок. При этом переменные имеют вид $x_{h(i)}$, $h(i)$ есть функция $i+m$ и m фиксированная для каждой обобщенной связки, положительная или отрицательная целая константа. Из этого ограничения вытекает, что если формулы F_i не содержат обобщенных связок, то они изоморфны.

Пусть $G = \cup_{i=k,h}\ F_i(x_{h(i)})$ $(G = \cap_{i=k,h}\ F_i(x_{h(i)}))$ есть формула, в которой $x_{h(i)}$ переменная. Тогда говорим, что переменная $x_{h(i)}$ управляется обобщенной логической связкой $\cup_{i=k,h}$ $(\cap_{i=k,h})$. Если переменная $x_{h(i)}$ не имеет индекса вообще, или функция h не зависит от i, то говорим, что эта переменная свободна от индекса i. Последний случай имеет место, когда переменная находится в области действия обобщенной связки, но ее индекс не связан с индексом связки.

Введем аналогичные понятия для обобщенных связок. Пусть $G = \cup_{i=k,h}\ F_i(H)$ $(G = \cap_{i=k,h}\ F_i(H))$ есть формула, в которой H есть собственная подформула вида $\cap_{j=s,p}N_j$ или $\cup_{j=s,p}N_j$, причем верхний или нижний пределы s или p есть функции, зависящие от i. Тогда говорим, что подформула H управляется связкой $\cup_{i=k,h}$ $(\cap_{i=k,h})$. Если оба предела s, p не зависят от i, то говорим, что связка $\cap_{j=s,p}$ или $\cup_{j=s,p}$ свободна от i.

Следующее ограничение определяет вид верхних и нижних границ обобщенных связок: это могут быть константы, символ бесконечности или выражения зависящие от индексов управляющих обобщенных связок. В последнем случае выполняются следующие ограничения на вид функций, определяющих значения границ: это могут быть лишь выражения вида $i+m$, где i - индекс обобщенной связки и m - некоторое целое (положительное или отрицательное) число, фиксированное для каждой конкретной обобщенной связки.

Подформула называется свободной от индекса i, если ни среди индексов ее переменных, ни среди верхних и нижних границ ее обобщенных связок не имеет места существенная

зависимость от i. Справедливо утверждение, что *каждая обобщенная формула эквивалентно приводима к виду, когда всякая ее переменная, свободная от индекса какой-либо обобщенной связки, не находится в ее области действия.*

Оказывается, что обобщенные формулы эквивалентно приводимы к некоторому достаточно простому виду, который позволяет судить о выразительных возможностях этого класса формул.

Пусть F и H суть обобщенные логические функции, $V = Var(F) \cup Var(H)$. Тогда эти функции называются *д-эквивалентными* относительно переменных множества V, если множества проекций единичных означиваний обеих функций на переменные V совпадают.

Обобщенные формулы обладают одним интересным свойством, которое назовем *локальностью*. В частности назовем формулу локальной, если хотя бы при одном упорядочении ее переменных любые означивания начальной относительно этого порядка последовательности переменных, разбиваются на классы эквивалентности, число которых конечно и определяется лишь видом самой формулы. Под эквивалентностью наборов, означивающих одно множество логических переменных, здесь понимается следующее отношение: два набора эквивалентны тогда и только тогда, когда при подстановке соответствующих значений вместо переменных порождаются логически эквивалентные функции.

Удается показать, что *всякая обобщенная формула д-эквивалентна некоторой локальной формуле.*

Это синтаксическое ограничение позволяет исследовать выразительные возможности обобщенных формул.

Логическая формула $F(x_1, x_2, \ldots, x_n, x_{n+1})$ представляет функцию $f(x_1, x_2, \ldots, x_n)$ двоичной арифметики, если при означивании переменных $x_1, x_2, \ldots, x_n, x_{n+1}$ бинарными наборами соответственно $s_1, s_2, \ldots, s_n, s_{n+1}$ такими, что $s_1, s_2, \ldots, s_n, s_{n+1}$, представляют двоичные числа, формула $F(s_1, s_2, \ldots, s_n, s_{n+1})$ истинна тогда и только тогда, когда $f(s_1, s_2, \ldots, s_n) = s_{n+1}$. Отметим, что формула $F(x_1, x_2, \ldots, x_n, x_{n+1})$ может содержать и другие переменные, кроме указанных.

Легко показать, что если формула представляет функцию двоичной арифметики, то любая д-эквивалентная ей формула также представляет эту функцию.

Пример. Логическая формула

$$\text{Сл1}(\mathbf{x}, \mathbf{z})\text{: } (z_0 = x_0)\ \&\ \cap_{i=1,\omega}(z_i = x_i \oplus \cap_{j=0,i-1} x_j)$$

представляет двоичную функцию прибавления единицы.

Не всякая функция двоичной арифметики представима обобщенной формулой. В частности, никакой обобщенной формулой не представимо двоичное целочисленное умножение.

Еще одним представлением обобщенных формул служат конечные и бесконечные пропозициональные матрицы, которые мыслятся как множества векторов, каждый вектор есть в точности один столбец матрицы. Различаются два типа матриц: D-матрицы, соответствующие д.н.ф., и K-матрицы - к.н.ф.

Теперь из определения локальной формулы вытекает, что всякая обобщенная формула представляется так называемой ленточной K-матрицей с лентой, ширина которой зависит только от вида формулы. Здесь под ленточными матрицами понимаются такие, значащие символы в которых сгруппированы в районе главной диагонали. Отсюда вытекает, что *всякая матрица, представляющая арифметическую функцию, также имеет вид диагональной с конечной шириной ленты.*

На множестве матриц определены умножение и разность. Эти операции позволяют наглядно связывать вид формулы и ее единичных покрытий. В частности, удается показать, что *всякая выполнимая обобщенная формула обладает единичным означиванием, имеющим вид почти периодической последовательности.* Под почти периодической понимается последовательность с некоторым начальным не периодическим отрезком конечной длины, после которого следует бесконечная периодическая последовательность.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 01-01-00930)

Обучающая система по геометрии

С. В. Попов, Е. Е.Трифонова (Москва)

Чтобы стимулировать интерес школьников к учебе и сделать обучение более эффективным, создаются разнообразные компьютерные обучающие программы. Однако большинство из них представляют собой лишь красочные версии учебников с удобной навигацией. Авторы предлагают иной подход к созданию обучающих систем, исходя из приведенных ниже допущений.

Процесс усвоения знаний должен напоминать игру, чтобы работа с обучающей системой стимулировала мотивацию достижения новых результатов. Поэтому система должна уметь решать задачи, которые не были введены и заранее решены ее создателями.

Если реализация первого из высказанных принципов не составляет особого труда, то при воплощении второго возникают существенные сложности. Как научить программу решать практически любые задачи, из данной предметной области, алгоритм решения которых заранее неизвестен? Для решения этой проблемы предлагается использовать подход, получивший название *логического моделирования.* Суть его состоит в следующем.

Каждая предметная область обладает конечным набором основных понятий, между которыми существуют определенные отношения. Поэтому она представима в виде логического исчисления, сигнатура которого определяется ее содержательным словарем. Исчисление описывает средствами логико-математического языка необходимые понятия предметной области. Если же в него включить соотношения исходной задачи, то получим полную спецификацию задачи. Из практических соображений, можно полагать, что итоговая спецификация представляет собой исчисление первого порядка.

Метод решения задачи, заданной своей спецификацией, основывается на следующем положении: *модель логического исчисления, представляющего собой формальную спецификацию задачи, в качестве фрагмента включает ее решение.* Действительно, логическая модель спецификации, представляет собой совокупность отношений, которые превращают ее в истинное утверждение. Поэтому концепция логического моделирования сводит поиск решения задачи, заданной формальной спецификацией, к построению ее модели.

Построение логической модели основывается на следующем утверждении: *при определенных условиях, логическое исчисление само является программой, которая порождает его собственную модель.* Доказательство этого утверждения основывается на том, что всякая логическая формула обладает так называемой операционной семантикой, которая позволяет по формуле находить ее модель. В результате находится логическая модель спецификации.

Построение модели подразумевает конструирование так называемой неподвижной точки программы, однозначно соответствующей спецификации задачи. Доказано, что *всякая неподвижная точка является логической моделью спецификации.* Эти принципы воплощены в разрабатываемой авторами обучающей системе по планиметрии. Геометрия выбрана в качестве предметной области потому, что, во-первых, обладает сравнительно небольшим понятийным базисом, и во-вторых, при решении задач необходимы знания из смежных областей (тригонометрии и алгебры).

Уже работающая обучающая система позволяет решать достаточно интересные задачи пока без привлечения алгебраических и тригонометрических фактов. Поиск решения осуществляется совместно пользователем и системой. Последняя при этом выступает в роли контролера сделанных шагов и подсказчика новых идей. Для этого она обладает развитым графическим интерфейсом, позволяющим вводить и редактировать разнообразные задачи. Общение с системой носит характер диалога, когда пользователь высказывает гипотезы, а компьютер их принимает или отвергает. Тем самым, она позволяет решать задач, которые формулируются обучающимся. Предполагается, что найденное решение будет поясняться в содержательных терминах и при желании может быть сохранено и использовано при решении новых задач.

Укажем некоторые особенности предлагаемого подхода к разработке обучающей системы по геометрии.

Во-первых, геометрический редактор позволяет вводить любые планиметрические чертежи. Редактор определяет все необходимее соотношения между элементами чертежа, которые в последующем используются для решения. Поэтому фиксируются все углы, пересечения и коллинеарность отрезков, принадлежности точек тем или иным фигурам, равенства углов с коллинеарными сторонами и т.п. Некоторые отношения, такие как коллинеарность и равенство, являются эквивалентностями. Поэтому для них формируются соответствующие классы эквивалентности. В частности, для равных углов создаются классы равных углов и в каждом из них выделяется некоторый так называемый минимальный угол с кратчайшими отрезками, образующими его стороны. В последующем при работе с углами используются только минимальные углы.

Во-вторых, особое внимание уделяется работе с именами объектов: отрезков, углов, треугольников и т.п. Все имена упорядочены определенным образом, в частности, отрезки именуются двумя буквами, первой идет предшествующая в лексикографическом порядке; углы именуются, как обычно, тремя буквами и их имена также упорядочены и т.д. С точки зрения эффективности вычисления это существенно, так как не позволяет порождать различные имена для одинаковых объектов. В итоге, порождается только один треугольник ABC, и не порождаются треугольники BAC, CBA, BCA.

В-третьих, решение каждой задачи сводится к построению нормальной модели для полной спецификации задачи, задаваемой начальными условиями и описанием геометрических понятий. Нормальная модель имеет в качестве элементов носителя классы эквивалентности, а не единичные объекты. Это вполне согласуется с содержательными решениями задач, так как обычно на чертеже выделяются классы равных и параллельных отрезков, равных и подобных треугольников и т.д.

Наконец, интеллектуальным ядром системы является программа на языке логического моделирования Покос. Этот язык существенно отличается от обычных языков программи-

рования следующим. Его главная особенность состоит в его двухуровневости: первый или логический уровень предназначен для описания используемых понятий предметной области и соотношений между ними. В нашем случае такими понятиями служат: различные типы пересечения отрезков, треугольники, перпендикулярность, равенство фигур и т.п.

Второй уровень служит для описания переборных стратегий, которые используются при решении интеллектуальных задач. Для достаточно сложных интеллектуальных задач, пример которой здесь представляется, поиск неподвижной точки, как искомого решения, невозможно представить в виде детерминированного алгоритма, реализуемого обычной программой приемлемой сложности. В отличие от операторных языков программирования второй уровень языка Покос лишь фиксирует некоторую стратегию поиска, реализация которой приводит к построению логической модели исходной спецификации, т.е. к решению задачи. Под стратегией здесь понимается обобщенный план решения, а не само решение. Поэтому трудозатраты на разработку интеллектуальных систем средствами Покоса существенно ниже, чем при их реализации традиционными средствами программирования.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 01-01-00930)

Некоторые верхние оценки длин единичных проверяющих тестов относительно транспозиций переменных булевых функций

Д. С. Романов (Москва)

Пусть $f(x_1, x_2, \ldots, x_n)$ — булева функция, а $f_{(i,j)}(x_1, x_2, \ldots, x_n)$ — функция, полученная из $f(x_1, x_2, \ldots, x_n)$ транспозицией переменных x_i и x_j $(1 \le i < j \le n)$, то есть

$$f_{(i,j)}(x_1, x_2, \ldots, x_i, \ldots, x_j, \ldots, x_n) = f(x_1, x_2, \ldots, x_j, \ldots, x_i, \ldots, x_n).$$

Обозначим через W_f множество функций $\{f, f_{(i,j)} \mid 1 \le i < j \le n\}$. Множество T наборов значений переменных $x_1, x_2, \ldots, x_i, \ldots, x_j, \ldots, x_n$ называется единичным проверяющим тестом относительно транспозиций переменных функции f тогда и только тогда, когда для любой пары $(f, f_{(i,j)})$ неравных функций из W_f найдется набор $\tilde{\alpha}$ из T такой, что $f(\tilde{\alpha}) \neq f_{(i,j)}(\tilde{\alpha})$. Количество наборов в тесте T называется его длиной и обозначается через $L(T)$. Тест минимальной длины называется минимальным. Обозначим через $L^{\text{п}}(f(x_1, x_2, \ldots, x_n))$ длину минимального единичного проверяющего теста относительно транспозиций переменных для $f(x_1, x_2, \ldots, x_n)$.

Пусть Q — класс булевых функций. Тогда $Q(n)$ — множество всех булевых функций из класса Q, зависящих формально от фиксированных n переменных $x_1, x_2, \ldots, x_n$. Через $L^{\text{п}}(Q(n))$ обозначим функцию Шеннона длины единичного проверяющего теста относительно транспозиций переменных функций из класса Q, то есть функцию

$$L^{\text{п}}(Q(n)) = \max_{f(x_1,x_2,\ldots,x_n) \in Q(n)} L^{\text{п}}(f(x_1, x_2, \ldots, x_n)).$$

В работе [1] было показано, что для некоторой константы c имеет место оценка

$$L^{\text{п}}(P_2(n)) \ge c \cdot n \log_2 n.$$

В настоящей работе предлагаются нетривиальные верхние оценки длин минимальных единичных проверяющих тестов относительно транспозиций переменных булевых функций из некоторых классов.

Заметим, что если $f_{(i,j)}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n)$ и $f_{(j,k)}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n)$, то и $f_{(i,k)}(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n)$. Поэтому множество переменных $x_1, x_2, \ldots, x_n$ может быть разбито на непересекающиеся максимальные по включению переменных множества такие, что любая транспозиция различных переменных функции f, лежащих в одном множестве, не обнаруживается. Каждое такое множество назовем множеством симметричности функции f.

Обозначим через $A_t(n)$ (через $B_t(n)$) класс булевых функций $f(x_1, x_2, \ldots, x_n)$, у каждой из которых мощность каждого множества симметричности не более t (соответственно, не менее t).

Утверждение. *При $n \to \infty$, $t = t(n) = O(\sqrt{n})$ и $s = s(n)$ таком, что $\sqrt{n} = O(s)$, имеют место порядковые оценки:*

$$L^{\text{п}}(A_t(n)) = O(n\sqrt{n}), \qquad L^{\text{п}}(B_s(n)) = O(n\sqrt{n}).$$

Автор выражает благодарность профессору С. А. Ложкину за внимание к работе.

Работа поддержана грантами РФФИ N 02-01-01110 и N 01-01-00266.

Литература

1. Глазунов Н. И., Горяшко А. П. Об оценках длин обнаруживающих тестов для классов неконстантных неисправностей входов комбинационных схем // Изв. АН СССР. Серия "Техническая кибернетика". — 1986. — N 3. — С. 197-200.

Классы отношений объектов и взаимодействия объектов

В. С. Рублев, В. О. Дерябин, М. А. Иоссель
А. В. Карповский, Д. И. Лобачев, А. Р. Юсупов (Ярославль)

Одним из направлений развития технологии обработки баз данных является создание объектных систем. Обобщением этого опыта является стандарт объектных данных, разработанный *Группой Управления Объектными Данными ODMG* (см. [1]). Однако этот стандарт, основанный на объектно-ориентированном подходе, наряду с несомненными достоинствами имеет ряд недостатков.

Так, с одной стороны, одним из достоинств упомянутого стандарта является введение в описание класса отношений с объектами других классов (в описании последних требуется введение обратного отношения). Но, с другой стороны, такой способ приводит к описанию только бинарных отношений объектов. И хотя отношение более чем двух объектов можно заменить некоторым количеством бинарных отношений, но такая замена не эквивалентна и приводит к дополнительным трудностям при использовании. Далее, даже бинарные отношения объектов часто необходимо снабжать дополнительными характеристиками (информацией), которые в указанных стандартом конструкциях отсутствуют (и непонятно, где их описывать). Например, смесь как объект состоит из ингредиентов (тоже объектов), каждый из которых входит в смесь в некотором количестве, и потому заданием лишь отношений *смесь-ингредиент* нельзя описать количественные характеристики.

Указанные трудности предлагается преодолеть естественным образом: введением *классов отношений объектов*. И хотя определение таких классов не противоречит объектно-ориентированной парадигме (может быть задано традиционным способом при помощи множественного наследования классов, входящих в отношение), но важность введения классов отношений (позволяющих описывать сложные системы объектов и их отношений) настолько велика, что предлагается расширить объектно-ориентированную парадигму, дополнив ее классами отношений, а также неявными связями между обычным классом и классом отношения, в который входит этот класс.

Особую роль в системе с введенными классами отношений объектов играют методы этих классов. Поскольку каждый из таких методов описывает те или иные действия, связанные с отношением объектов и определяющих изменение информации, то введем для них термин: *взаимодействие объектов*. Каждый из методов обычных классов, связанный с созданием объектов, их коррекцией или удалением, также можно рассматривать как взаимодействие пользователя системы (как объекта) с информационным объектом класса.

При таком подходе взаимодействие становится основным действием в объектной системе, и при введении прав пользователя на взаимодействия его интерфейс включает меню возможных взаимодействий.

Отметим, что другие недостатки упомянутого стандарта (отсутствие множественного и внутреннего наследования, отсутствие динамики изменения свойств классов, объектов и их параметров) преодолеваются введением специфических базовых отношений объектной системы, рассмотренных в [2]-[4].

Литература

1. Cattel R.G.G, Barry D.K. and others. The Object Data Standart: ODMG 3.0. Morgan Kaufmann. January 2000.

2. Дерябин В. О., Рублев В. С. Принципы организации СУБД-независимого языка ALODS для объектно-динамических систем // Современные проблемы математики и информатики. // Сборник научных трудов молодых ученых, аспирантов и студентов. вып.2. Ярославский госуниверситет. Ярославль, 2000. — С.183-189.

3. Дерябин В. О., Лобачев Д. И., Рублев В. С., Юсупов А. Р. Базовые отношения объектов динамической информационной модели и гибкие таблицы данных // Проблемы теоретической кибернетики. Тезисы докладов XIII Международной конференции (Казань, 27–31 мая 2002 г.). Часть I. — М.:изд-во механико-математического факультета МГУ, 2002. — С.55.

4. Рублев В. С., Дерябин В. О., Лобачев Д. И., Юсупов А. Р. Базовые отношения обектов баз данных и гибкие таблицы //Моделирование и анализ информационных систем. Ярославль, 2002. Т.9, N 2. С.16-27.

О проверке на простоту чисел вида $N = 2kp^m - 1$

Е. В. Садовник (Москва)

Одной из важнейших криптографических задач является проверка чисел на простоту. На сегоднешний день существует алгоритм, позволяющий опредилить простоту числа N за время $O((\log N)^{12})$. Задача упрощается, если известно разложение на множители чисел

$N+1$, $N-1$ и т.п. В данной работе приводится алгоритм, позволяющий за время $O(\log N)$ опредилить простоту числа $N = 2kp^m - 1$, где p - простое число больше 2, m - натуральное число, k - натуральное, нечетное и $2k < p^m$.

Для того, чтобы сформулировать результаты, введем необходимые определения. Пусть P и Q – два взаимно простых числа, а α и β – корни квадратного уравнения $x^2 - Px + Q = 0$. Функции Люка определяются формулами

$$V_n(P,Q) = \alpha^n + \beta^n,$$

$$U_n(P,Q) = (\alpha^n - \beta^n)/(\alpha - \beta).$$

Пусть $\Delta = (\alpha - \beta)^2 = P^2 - 4Q$ – дискриминант квадратного уравнения. Функции Люка удовлетворяют следующим свойствам:

$$2V_{n+1} = PV_n + \Delta U_n, \qquad 2U_{n+1} = V_n + PU_n,$$

$$V_{2n} = V_n^2 - 2Q^n, \qquad U_{2n} = U_n V_n.$$

С помощью этих свойств можно быстро вычислять значения U_n и V_n для больших значений n. Эти и другие тождества можно найти в [1].

Сначали приведем алгоритм для случая когда $p = 3$.

Теорема 1. *Пусть $N = 2k3^m - 1$, где m - натурально чило, k - натуральное нечетное число и $2k < 3^m$. Обозначим $S_1 = V_k(-14, 1)$ и*

$$S_{i+1} = S_i\left(S_i^2 - 3\right) \pmod N,$$

Тогда если $S_m \neq -2 \pmod N$, то N - простое тогда и только тогда, когда $N|(S_m^2 - 1)$.

По этому алгоритму была написана программа, которая показала, что на промежутке от $m = 1$ до 14, во всех случаях, когда $S_m = -2 \pmod N$, числа N были простыми, кроме $N = 2 \cdot 55 \cdot 3^6 - 1$.

Для произвольного простого p алгоритм будет следующим.

Теорема 2. *Пусть $N = 2kp^m - 1$, где p - простое число и $p = 3 \pmod 4$, m - натурально чило, k - натуральное нечетное число и $2k < p^m$. Пусть $Q = \pm p$ и $P^2 = p(d^2 \pm 4)$ для некоторого натурального d. Тогда, если $V_{(N+1)/2p}(P,Q) \neq 0 \pmod N$, то N - простое тогда и только тогда, когда* нод $\left(U_{(N+1)/p}(P,Q), N\right) = 1$ *и $N|U_{(N+1)}(P,Q)$.*

Доказана также теорема 3, являющаяся обобщением теоремы 2.

Теорема 3. *Пусть $N = 2 \cdot k \cdot p_1^{m_1} \cdot \ldots \cdot p_l^{m_l} - 1$, где p_i - простые числа и $p_1 \cdot \ldots \cdot p_l = 3 \pmod 4$, m_i - натуральные чила $\forall i = 1, \ldots, l$, k - натуральное нечетное число и $2k < p_1^{m_1} \ldots p_l^{m_l}$. Пусть $Q = \pm p_1 \cdot \ldots \cdot p_l$ и $P^2 = Q(d^2 \pm 4)$ для некоторого натурального d. Тогда, если $V_{(N+1)/2p_i}(P,Q) \neq 0 \pmod N$, то N - простое тогда и только тогда, когда* нод $\left(U_{(N+1)/p_i}(P,Q), N\right) = 1$ $\forall i = 1, \ldots, l$ *и $N|U_{(N+1)}(P,Q)$.*

Литература

1. Уильямс Х. Проверка чисел на простоту с помощью вычислительных машин // Кибернетический сборник, вып. 23, М., Мир, 1986, с. 51-99.

Доказательство гипотезы Камерона-Эрдеша о числе множеств, свободных от сумм

А. А. Сапоженко (Москва)

Подмножество A целых чисел называется *свободным от сумм*, (сокращенно, *MCC*) если для любых $a, b \in A$ число $a+b$ не принадлежит множеству A. Для любых действительных чисел q и p обозначим через $[q,p]$ множество натуральных чисел x таких, что $q \le x \le p$. Семейство всех подмножеств $A \subseteq [t,n]$, свободных от сумм, обозначим через $S(t,n)$. Пусть $s(t,n) = |S(t,n)|$, а $s(n) = |S(1,n)|$. В 1988 г. П. Камерон и П.Эрдеш [1] предположили, что $s(n) = O(2^{n/2})$. Кроме того они доказали, что $s(n/3,n) = O(2^{n/2})$. Целью данного сообщения является доказательство гипотезы Эрдеша-Камерона.

Идея доказательства. Мы сводим задачу об оценке $s(n)$ к оценке числа независимых множеств в графе Кэли. Если $F \subseteq [1,n]$ и $V \subseteq [1,n]$, то граф $\mathcal{C}_F(V)$ с множеством вершин V, в котором пара $\{i,j\} \subseteq V$ является ребром тогда и только тогда, когда $|i-j| \in F$ или $i+j \in F$ называется *графом Кэли* на множестве V относительно F.

Пусть $l \le k-\theta \le k+\theta \le m$. Граф с n вершинами, в котором минимальная степень вершины равна l, максимальная степень вершины равна m, доля вершин, степень которых больше $k+\theta$, равна Δ, доля вершин, степень которых меньше $k-\theta$, равна δ, назовем $(n,l,k,m,\delta,\Delta,\theta)$-*графом*. Подмножество A вершин графа G называется *независимым*, если подграф, порожденный множеством A, не содержит ребер. Число всех независимых множеств графа G обозначим через $I(G)$. Семейство подмножеств $\mathcal{F}$ вершин графа G назовем *покрывающим*, если для любого независимого множества A графа G существует $D \in \mathcal{F}$, такое, что $A \subseteq D$. Доказательство опирается на следующие факты.

Теорема 1 (см. [2]). *Пусть $G = (V,E)$ является $(n,l,k,m,\delta,\Delta,\theta)$-графом. Тогда существует покрывающее семейство $\mathcal{F}$, удовлетворяющее следующим условиям:*

1) *для всякого $D \in \mathcal{F}$*

$$|D| \le \frac{n}{2}\left(1+\delta(1-l/k)+\Delta(m/k-1)+O((\theta+\sqrt{k\log k}\,)/k)\right); \qquad (1)$$

$$2) \qquad |\mathcal{F}| \le 2^{n\sqrt{\frac{\log k}{k}}}. \qquad (2)$$

Другим фактом является утверждение о структуре MCC, имеющих плотное подмножество, состоящее из начальных элементов отрезка $[1,n]$ (см. тезисы К.Г.Омельянова и А.А.Сапоженко).

Элементы покрывающего семейства $\mathcal{F}$ назовем контейнерами. *Фрагментом* контейнера $D \in \mathcal{F}$ называется множество вида $D \cap [k,l]$. Семейство контейнеров называется *правильным*, если "достаточно крупные" фрагменты каждого контейнера является правильным, т.е. содержат приблизительно половину четных и половину нечетных чисел, а разность наибольшего и наименьшего из чисел фрагмента приблизительно в два раза больше, чем его мощность. Доказывается, что все MCC за исключением подмножеств A, состоящих из четных чисел, а также MCC, являющихся подмножествами множества $[n/3,n]$, содержатся в элементах некоторого правильного семейства контейнеров. Основной результат следует из того, что число MCC, содержащихся в элементах правильного семейства контейнеров есть $o(2^{n/2})$.

Работа выполнена при поддержке РФФИ (проект 01-01-00266).

Литература

1. Cameron P., Erdos P., On the number of integers with various properties// in R. A. Mollin (ed). Number Theory: Proc. First Conf. Can. Number Th. Ass., Banff, 1988, — de Gruyter. 1990 — P. 61-79.

2. Сапоженко А.А., О числе независимых множеств в графах// Труды VIII международной конференции по проблемам теоретической кибернетики (Казань, 27-31 мая 2002 г.) (в печати).

О сложности поляризованных полиномов функций k-значных логик, зависящих от одной переменной

С. Н. Селезнева (Москва)

В настоящей заметке рассматриваются полиномы, в которых каждая переменная может быть поляризована смещением на определенную величину. Такие полиномы называются поляризованными. Вводится функция Шеннона сложности задания функций многозначных логик поляризованными полиномами и находятся некоторые ее оценки.

Пусть $k \geq 3$, $E_k = \{0, 1, \ldots, k-1\}$. Функцией k-значной логики называется отображение $f^n : E_k^n \to E_k$, $n = 0, 1, \ldots$. Множество всех функций k-значной логики обозначим через P_k.

Пусть число k – простое. Тогда каждая функция k-значной логики может быть записана (причем однозначно) полиномом относительно операций сложения и умножения по $\mathrm{mod}\ k$. Обозначим полином, задающий функцию $f(x_1, \ldots, x_n)$, через $P(f)$.

Пусть $\delta = (d_1, \ldots, d_n)$, где $d_1, \ldots, d_n \in E_k$. *Поляризованным полиномом* функции $f(x_1, \ldots, x_n)$ по *вектору поляризации* δ назовем полином по $\mathrm{mod}\ k$ функции $f(x_1 - d_1, \ldots, x_n - d_n)$ и обозначим его $P^\delta(f)$. Из существования и однозначности полинома, задающего функцию k-значной логики при простом k, следует, что для каждой функции k-значной логики поляризованный полином по каждому вектору поляризации δ существует и однозначен.

Будем полагать, что в рассматриваемых полиномах приведены все подобные слагаемые. Назовем длиной полинома P число его слагаемых с ненулевыми коэффициентами и обозначим ее $l(P)$. Введем понятие сложности функции $f(x_1, \ldots, x_n)$ при задании ее поляризованными полиномами: $l(f) = \min\ l(P^\delta(f))$, где минимум берется по всем возможным векторам поляризации δ.

Для каждого простого k и всех n, $n \geq 1$, рассмотрим функцию Шеннона $L_k(n)$ сложности представления функций k-значной логики, зависящих от n переменных, поляризованными полиномами: $L_k(n) = \max\ l(f)$, где максимум берется по всем функциям f из P_k, зависящим от n переменных.

Перязевым Н. А. в [1] была получена точная оценка функции Шеннона для булевых функций:

$$L_2(n) = [\frac{2}{3} \cdot 2^n],$$

где выражение $[a]$ обозначает целую часть числа a. В [2] автором была доказана верхняя оценка функции Шеннона:

$$L_k(n) \leq \frac{k(k-1)}{k(k-1)+1} \cdot k^n.$$

В настоящей заметке дана точная оценка функции Шеннона для функций, зависящих от одной переменной.

Теорема 1. *Если k – простое, то $L_k(1) = k - 1$.*

Следствие. *Если k – простое, то $L_k(n) \geq (k-1)^n$*

Работа поддержана грантом РФФИ, код проекта 00-01-00351.

Литература

1. Перязев Н. А. Сложность булевых функций в классе полиномиальных поляризованных форм. Алгебра и логика, 34 (1995), N 3, 323-326.

2. Селезнева С. Н. О сложности представления функций многозначных логик поляризированными полиномами. Дискретная математика, т. 14 (2001), вып. 2, 48-53.

Построение систем определяющих соотношений для автомата

А. С. Сенченко (Славянск)

Рассматриваются конечные всюду определнные, инициальные связные автоматы $A = (A, X, \delta, a_0)$, у которых A – множество состояний, X – множество входных символов, $\delta : A \times X \to A$ – функция переходов обычным образом расширенная на множество X^* входных слов конечной длины. Полагаем $\delta(a, p) = a \cdot p$ для $a \in A$, $p \in X^*$. Пусть ρ_A – правая конгруенция [1] на X^*, определенная правилом: $(p, q) \in \rho_A$, если $a_0 p = a_0 q$. Ясно, что $X^*/\rho_A = A$. Конечное отношение ρ на X^* назовем системой определяющих соотношений (СОС) для A, если правоконгруентное замыкание $[\rho]$ равно ρ_A.

Для каждого состояния a_i находим кратчайшее в лексикографическом порядке $\leq$ слово p_i, для которого $a_0 p_i = a_i$. Множество V таких слов является базисом достижимости A [2]. Для каждого $p_i x \in V \cdot X$, которое не входит в V образуем пару (p_i, p_j), где $a_0 p_i x = a_0 p_j$ и $p_j \in V$. Все такие пары образуют конечное бинарное отношение κ_A. Назовем его каноническим.

Пусть ρ – произвольное конечное бинарное отношение на X^*. Проведем его преобразование, состоящее из следующих операций:

1.Удалим из ρ пары вида (p, p).

2. Пару $(p, q) \in \rho$, если $q < p$ замени парой (q, p).

3. Пусть $(p, q), (t, u) \in \rho$. Если $p = uw$ для некоторого $w \in X^*$, то пару (p, q) в ρ заменяем парой (tw, q). Если же $q = uw$, то пару (p, q) в ρ заменяем парой (p, tw).

4. Удалим повторяющиеся пары, оставляя только по одному экземпляру.

Рассмотрим процедуру редукции отношения ρ, состоящую из таких шагов:

а) выполняется операция 1;

б) выполняется операция 2;

в) если операция 3 неприменима, то процедура завершается, иначе после каждого выполнения операции 3 переходим к операции 4;
г) выполняется операция 4 и переходим к (а).

Полученное бинарное отношение обозначим $\langle\rho\rangle$.

Теорема 1. *Отношение ρ является СОС для A тогда и только тогда, когда $\langle\rho\rangle = \kappa_A$.*

Доказательство теоремы проводится в [3].

Из теоремы 1. вытекают следствия.

Следствие 1.1. Пусть ρ - конечная система определяющих соотношений для автомата A, а ρ'- некоторое конечное бинарное отношение на множестве X^*. ρ' будет системой определяющих соотношений для A тогда и только тогда, когда $\langle\rho'\rangle = \langle\rho\rangle$.

Следствие 1.2. κ_A - минимальная система определяющих соотношений для автомата A по количеству пар и по суммарной длине слов.

Конечное множество $\Omega = \{w_1, \ldots, w_k\}$ слов в алфавите X назовем обходом автомата $A \in K(X)$, если для любых $a \in A$ и $x \in X$ найдется такое слово w', что $a_0 w' = a$, и слово $w'x$ является начальным отрезком некоторого слова $w_i \in \Omega$.

Рассматривается взаимосвязь между обходами автомата и его СОС.

Теорема 2. *Пусть ω – некоторый обход по дугам автомата A, $W = \{e = w_0, w_1, ..., w_k\}$ – множество начальных отрезков обхода ω, $F_A = \{f_0, f_1, ..., f_{n-1}\}$ – произвольный базис достижимости автомата A. Пусть $\rho' = \{(f_i, w_j)\}$, $j = 0, ..., k$ такое, что $w_j \in W$, $f_i \in F_A$ и $a_0 f_i = a_0 w_j$. Тогда ρ' является СОС для A.*

Обход $\Omega = \{\omega_1, \ldots, \omega_k\}$ автомата $A \in K(X)$ назовем избыточным, если $\omega_i = \omega_j z \quad (i \neq j)$, для некоторых слов $\omega_i, \omega_j \in \Omega$, $z \in X^*$, в противном случае обход считаем неизбыточным.

Обозначим через O_A множество неизбыточных обходов автомата $A \in K(X)$, а через C_A множество его систем определяющих соотношений.

Введем отображения $f : O_A \to C_A$ и $g : C_A \to O_A$.

Пусть $\Omega \in O_A$, W_Ω - множество всех начальных отрезков обхода Ω, V_Ω - кратчайший по лексикографическому порядку $\leq$ внутренний базис достижимости автомата A относительно Ω. Пусть $\lambda = \{(v_i, w_j)\}$ - множество всех таких пар слов, что $a_0 v_i = a_0 w_j$, $v_i \in V_\Omega$, $w_j \in W_\Omega$. Затем удалим из λ все пары вида (p, p). Полученное отношение обозначим через τ. Положим $\Omega f = \tau$.

Пусть $\rho \in C_A$. Исключим из W_ρ все слова, являющиеся начальными отрезками других слов из W_ρ. Обозначим полученное множество через $]W_\rho[$. Положим $\rho g =]W_\rho[$.

Теорема 3. *Пусть $\Omega \in O_A$, $\rho \in C_A$. Тогда $\Omega f \in C_A$, $\rho g \in O_A$.*

Теорема 4. *f - инъекция, g - сюръекция , причем $\Omega f g = \Omega$ для любого неизбыточного обхода $\Omega \in O_A$. и $\Omega f g = \Omega$.*

Автор благодарит Грунского И.С. за постановку задачи и помощь в ее решении.

Литература

1. Богомолов А.М., Салий В.Н., Алгебраические основы теории дискретных систем. – М.: Наука. Физматгиз, 1997 – 368 с.

2. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов – М.: Наука, 1985. – 320 с.

3. Грунский И.С., Сенченко А.С. Каноническая система определяющих соотношений для автоматов // Труды ИПММ НАН Украины, 2002, т.7. – с. 58-63.

О построении корреляционно-иммунных и устойчивых булевых функций

Ю. В. Тарannиков (Москва)

Булева функция от n переменных — это отображение из F_2^n в F_2. *Вес* $\mathrm{wt}(f)$ функции f над F_2^n — это число наборов x из F_2^n, для которых $f(x) = 1$. Функция f называется *уравновешенной*, если $\mathrm{wt}(f) = \mathrm{wt}(f \oplus 1) = 2^{n-1}$. Для заданной функции f из F_2^n минимум расстояний от нее до множества всех аффинных функций из F_2^n называется *нелинейностью* функции f и обозначается через $\mathrm{nl}(f)$.

Булева функция f, заданная на F_2^n, называется *корреляционно-иммунной порядка* m, $1 \le m \le n$, $\mathrm{wt}(f') = \mathrm{wt}(f)/2^m$ для любой ее подфункции f' от $n - m$ переменных. Уравновешенная корреляционно-иммунная функция порядка m называется *m-устойчивой.*

Преобразованием Уолша булевой функции f называется целочисленная функция над F_2^n, определяемая следующим образом: $\widehat{\chi}_f(u) = \sum\limits_{x \in F_2^n} (-1)^{f(x)+<u,x>}$. Значения $\widehat{\chi}_f(u)$ называются *коэффициентами Уолша.*

Для коэффициентов Уолша справедливы
формула обращения: $(-1)^{f(x)} = 2^{-n} \sum\limits_{u \in F_2^n} \widehat{\chi}_f(u)(-1)^{<u,x>}$,
равенство Парасеваля: $\sum\limits_{u \in F_2^n} \widehat{\chi}_f^2(u) = 2^{2n}$,
а также *тождество Саркара:* $\sum\limits_{\substack{u \in F_2^n \\ u \preceq w}} \widehat{\chi}_f(u) = 2^n - 2^{|w|+1}\mathrm{wt}(f_w)$,
где $w \in F_2^n$, а f_w — функция, полученная из f подстановкой $0 \to x_i$ для всех таких i, что $w_i = 1$.

Хорошо известны следующие характеризации. Функция f на F_2^n является корреляционно-иммунной функцией порядка m тогда и только тогда, когда $\widehat{\chi}_f(w) = 0$ для всех наборов $w \in F_2^n$, для которых $1 \le |w| \le m$. Если f является корреляционно-иммунной функцией порядка m на F_2^n, $m \le n - 1$, то для любого $w \in F_2^n$ выполнено $\widehat{\chi}_f(w) \equiv 0 \pmod{2^{m+1}}$. Более того, если f является m-устойчивой или же $\widehat{\chi}_f(0) \equiv 0 \pmod{2^{m+2}}$, $m \le n - 2$, то $\widehat{\chi}_f(w) \equiv 0 \pmod{2^{m+2}}$.

Нелинейность функции f выражается формулой $\mathrm{nl}(f) = 2^{n-1} - \frac{1}{2} \max\limits_{u \in F_2^n} |\widehat{\chi}_f(u)|$. Автором и рядом других исследователей ранее было доказано, что

$$\mathrm{nl}(f) \le 2^{n-1} - 2^{m+1} \tag{1}$$

для m-устойчивых функций на F_2^n, причем равенство может достигаться только если $\widehat{\chi}_f(u) \in \{0, \pm 2^{m+2}\}$; кроме того,

$$\mathrm{nl}(f) \le 2^{n-1} - 2^m \tag{2}$$

для корреляционно-иммунных порядка m функций на F_2^n, причем равенство может достигаться только если $\widehat{\chi}_f(u) \in \{0, \pm 2^{m+1}\}$. Заметим, что при $m \le 0,5n - 2$ граница (1) достигаться не может. Автором ранее были построены алгебраические конструкции функций, достигающих границы (1) при всех парах (m, n), удовлетворяющих соотношениям $0,6n - 1 \le m \le n - 2$. Пасаликом, Майтрой, Йоханнсоном и Саркаром были построены функции, достигающие (1) при $n = 7$, $m = 2$. Недостижимость границы (2) не доказана

лишь для небольшой доли (хотя и бесконечного числа) пар (m, n). Тем не менее существует некоторые значения параметров, для которых достижимость границ (1) и (2) является открытой и практически актуальной проблемой. В частности, наименьшими параметрами, для которых проблема открыта, являются $n = 9$, $m = 3$ для границы (1) и $n = 9$, $m = 4$ для границы (2).

В настоящей работы автором разработаны алгоритмы построения устойчивых и корреляционно-иммунных булевых функций с высокой нелинейностью, использующие технику комбинаторной оптимизации. На первом этапе определяется на каких наборах коэффициенты Уолша будут нулевыми, а на каких — ненулевыми, так, чтобы не возникало противоречий в сравнимостях, получающихся из тождества Саркара. На втором этапе производится попытка расставить плюсы и минусы у ненулевых коэффициентов Уолша так, чтобы формулы обращения дали искомую булеву функцию. С помощью разработанных алгоритмов построены 2-устойчивые функции от 7 переменных, достигающие границы (1), у которых число ненулевых коэффициентов Уолша веса 3 принимает все возможные значения от 14 до 31 включительно (все ранее построенные функции с такими параметрами имели 15 таких коэффициентов). Вопрос существования и построения упомянутых выше функций при $n = 9$ остается пока открытым.

Работа поддержана грантами РФФИ 02-01-00985 и УР.04.03.007.

Литература

1. Тарannиков Ю. В. О корреляционно-иммунных и устойчивых булевых функциях, Математические вопросы кибернетики, Вып. 11, М., Физматлит, 2002, с. 91–148.

О классах многоэкстремальных функций, допускающих поиск глобального экстремума методом Фибоначчи

В. П. Тарасова (Москва)

Рассматриваются возможности эффективного применения известного метода Фибоначчи [1] к поиску глобального экстремума многоэкстремальных кусочно-непрерывных функций, и проблема описания классов таких функций. Используется подход, при котором решается обратная задача: ищется не метод, решающий задачу для заданного класса функций, а разыскивается класс функций, для которого заданный метод оказывается оптимальным методом поиска. Этот подход позволяет выявить область применимости метода Фибоначчи. В работе исследован класс функций, имеющих один глобальный максимум и любое число локальных экстремумов. Используется понятие отрезка *наибольших значений* P функции $f(x)$ [2, 3], определяемого условием: из $x_1 \in P$ и $x_2 \notin P$ следует $f(x_1) > f(x_2)$. Задается положительное число δ, и из множества M кусочно-непрерывных функций выделяется подмножество Q_δ функций, удовлетворяющих следующим условиям: каждая функция f из Q_δ имеет отрезок наибольших значений длины δ, является в этом отрезке унимодальной и произвольна вне его, то есть может иметь здесь любое число локальных экстремумов.

Из этого определения следует, что множеству Q_δ принадлежат одновременно все функции из M , имеющие отрезок наибольших значений длины большей, чем δ, а также заве-

домо принадлежат все унимодальные функции. Множество Q_δ является, в определенном смысле, расширением множества унимодальных функций U. В работе решены две задачи.
Задача 1. Указать такое δ, если оно существует, при котором для заданного натурального n, $n > 2$, n-шаговая стратегия Фибоначчи Φ_n решает задачу поиска точки глобального максимума для любой функции из множества Q_δ.
bf Задача 2. Указать все такие n, если они существуют, при которых для заданного положительного δ стратегия Φ_n решает задачу поиска точки глобального максимума для любой функции из множества Q_δ.

Решение задачи 1 дает

Лемма 1. *При заданном натуральном числе n, $n > 2$, стратегия Φ_n решает задачу поиска точки глобального максимума для любой функции из множества Q_δ тогда и только тогда, когда $\delta \geq |a,b|F_{n-2}/F_n$, где F_n – n-е число Фибоначчи, и $|a,b|$ - длина отрезка [a, b].*

Решение задачи 2. Значения функции $r(n) = F_{n-2}/Fn$ упорядочиваются в виде ряда

$$1/2, 2/5, \ldots, F_{2l-2}/F2l, F_{2l}/F_{2l+2}, \ldots, 0,38, \ldots, F_{2k+1}/F_{2k+3}, F_{2k-1}/F_{2k+1}, \ldots, 3/8, 1/3.$$

Выделяются соответствующие членам этого ряда множества

$$Q_{1/2}, Q_{2/5}, Q_{5/13}, \ldots, Q_{0,38}, \ldots, Q_{8/21}, Q_{3/8}, Q_{1/3},$$

для которых выполняются включения

$$Q_{1/2} \subset Q_{2/5} \subset Q_{5/13} \subset \ldots \subset Q_{0,38} \subset \ldots Q_{8/21} \subset Q_{3/8} \subset Q_{1/3}.$$

Лемма 2. *Для класса функций $Q_{1/3}$ стратегия Фn решает Задачу 2 при единственном значении $n = 3$.*

Для класса функций $Q - \alpha$, $\alpha = F_{2k-1}/F_{2k+1}$, $k = 1, 2, \ldots$, стратегия Φ_n решает Задачу 2 только при тех нечетных значениях n, которые меньше, или равны $2k+1$, $k = 1, 2, \ldots$.

Для класса функций Q_β, $\beta = F_{2l}/F_{2l+2}$, $l = 1, 2, \ldots$, стратегия Φ_n решает Задачу 2 при всех нечетных n, и только при тех четных n, которые больше или равны $2l + 2$, $l = 1, 2, \ldots$.

Для класса $Q_{1/2}$ функций стратегия Φ_n решает Задачу 2 при всех натуральных $n \geq 2$.

Через $\{Q_r\}$ обозначаем совокупность множеств

$$U, Q_{1/2}, Q_{2/5}, Q_{5/13}, \ldots, Q_{0,38}, \ldots, Q_{8/21}, Q_{3/8}, Q_{1/3}.$$

Теорема 1. *Семейством множеств $\{Q_r\}$ исчерпываются множества кусочно-непрерывных многоэкстремальных функций с единственным глобальным максимумом, для которых стратегия Фибоначчи Φ_n решает задачу поиска точки глобального экстремума.*

Теорема 2. *Если стратегия Φ_n решает задачу поиска точки глобального максимума для функций множества Q_δ, то она является ϵ-оптимальной (в минимаксном смысле) стратегией поиска точки глобального максимума для этого множества функций.*

Литература

1. Kiefer J. Sequential minimax search for a maximum // Proc. Amer. Math. Soc., 1953, N 3, p. 502-505.

2. Тарасова В.П. Метод стратегии противника в задачах оптимального поиска. М.: Изд-во МГУ, 1988.

3. Тарасова В.П. Оптимальный поиск экстремума для класса локально-унимодальных функций // Кибернетика,1984, 1, с.65-68.

Дискретное моделирование преобразования квантового бита

С. В. Шалагин (Казань)

Конфигурация квантовой системы (ККС), определенная в виде $|\psi\rangle = \left(z_1 e^{i\alpha}, z_2 e^{i\beta}\right)^T$, представима вектором Блоха в декартовой прямоугольной системе координат (ДПСК) $\vec{b} = \{s_x^2, s_y^2, s_z^2\}$ [1], где $s_x = 2|z_1||z_2|\cos(\alpha-\beta)$, $s_y = 2|p_1||p_2|\sin(\alpha-\beta)$, $s_z = 2|p_1|^2 - 1 = 1 - 2|p_2|^2$.

При переходе от ДПСК к сферической системе координат $\vec{b}$ определяем системой вида $(|\vec{b}| = 1, \theta, \phi)$, где θ - угол между осью Z и $\vec{b}$, ϕ - угол между осью X и проекцией $\vec{b}$ на XOY в ДПСК. Для определения $\vec{b}$ в ССК достаточно задать значения из множества $M = (\theta, \phi), \theta, \phi \in [0^o, 360^o)$.

При этом $\theta = \arccos(2|p_1|^2 - 1) = \arccos(1 - 2|p_2|^2)$, $\phi = \alpha - \beta$.

Обратный переход от представления $\vec{b}$ в ССК к представлению в ДПСК осуществляется по формулам [1]:

$$s_x = \sin\theta\cos\phi, \quad s_y = \sin\theta\sin\phi, \quad s_z = \cos\theta.$$

Опеделим дискретное множество значений $D = Li$, $L = \frac{360^o}{2^n-1}$, $i = \overline{0, 2^n - 2}$.

Определение. Величины $\theta, \phi = U$, $\theta, \phi \in D$, однозначно задающие конфигурацию квантовой системы, определяются элементами поля Галуа вида $GF(2^n)$ [2], где $\forall\lambda \in U$ соответствует элемент $\xi^i \in GF(2^n)$, для которого $i = [\xi/L + 1/2] mod(2^n - 1)$, а ξ - примитивный элемент $GF(2^n)$.

Операция по варьиванию ККС отображается путем изменения значений (θ, ϕ) на величины $\triangle\theta$ и $\triangle\phi$ соответственно, $\triangle\theta$, $\triangle\phi \in D$. Новая ККС определена как $(\theta + \triangle\theta, \phi + \triangle\phi)$. Изменение величин производится посредством набора квантовых гейтов вида $R_y(\tau)$ и $R_z(\gamma)$ [3]. Операция, направленная на изменение ККС, определена квантовым гейтом: $G = R_z(-\phi)R_y(-\triangle\theta)R_z(\phi + \triangle\phi), \phi, \triangle\phi, \triangle\theta \in D$.

В результате сформулирована

Теорема. *Операция по изменению конфигурации квантовой системы, описанная квантовым гейтом G, моделируется посредством двух операций умножения над элементами поля Галуа* $GF(2^n)$.

Работа выполнена при финансовой поддержке РФФИ, проект N 03.01.00769.

Литература

1. Валиев К.А., Кокин А.А. Квантовые компьютеры: надежды и реальность. - М.: Ижевск: R&C Dynamics, 2001.

2. Лидл Р., Нидеррайтер Г. Конечные поля: В 2 Т. - М.: Мир, 1988.

3. A.Barenco, C.H.Bennett, R.Cleve at all. Elementary gates for quantum computation // Physical Review A, March 22, 1995 (AC5710).

О выводе общезначимых формул из аксиом в логике ветвящегося времени

Р. В. Хелемендик (Москва)

Введение.

Система аксиом для логики ветвящегося времени предложена в [1]. Там же доказана и ее полнота без указания метода вывода общезначимых формул из аксиом. Отметим, что в этой работе на графы в моделях для формул было наложено ограничение тотальности: всякая вершина в графе имеет последователя. В общем случае, когда отсутствует ограничение тотальности, в [1] приведена система аксиом с указаниями сведения общего случая к частному. Этот общий случай рассматривается в настоящей работе, и описывается метод вывода всякой общезначимой формулы из аксиом, откуда, в частности, следует полнота системы аксиом.

Основные определения.

• Каждая пропозициональная переменная есть *формула*.

• Если φ, ψ *формулы*, то θ, являющаяся одним из 12 выражений $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$, $\neg\varphi$, $\forall\bigcirc\varphi$, $\exists\bigcirc\varphi$, $\forall\Box\varphi$, $\exists\Box\varphi$, $\forall\Diamond\varphi$, $\exists\Diamond\varphi$, $\forall(\varphi \mho \psi)$, $\forall(\varphi \mho \psi)$ тоже называется *формулой*.

• Других *формул* нет.

Модель - это пара $M = \langle \Gamma, L \rangle$, где Γ - это конечный связный ориентированный граф с выделенной вершиной u_0, а L - функция означивания, сопоставляющая каждой вершине множество пропозициональных переменных. *Полным путем* в графе называется бесконечный путь или цепь, последняя вершина которой не имеет сыновей.

Истинность формулы θ в вершине u_i модели M (обозначим это $M, u_i \models \theta$) определяется индуктивно.

• если θ есть пропозициональная переменная p, то $M, u_i \models \theta \iff p \in L(u_i)$

• если θ есть $(\varphi \wedge \psi)$ $[(\varphi \vee \psi)]$, то $M, u_i \models \theta \iff M, u_i \models \varphi$ и [или] $M, u_i \models \psi$

• если θ есть $(\varphi \to \psi)$, то $M, u_i \models \theta \iff M, u_i \models \psi$ или $M, u_i \not\models \varphi$ (неверно $M, u_i \models \varphi$)

• если θ есть $\neg\varphi$, то $M, u_i \models \theta \iff M, u_i \not\models \varphi$

• если θ есть $\forall\bigcirc\varphi$ $[\exists\bigcirc\varphi]$, то $M, u_i \models \theta \iff$ для каждого сына u_{i+1} (если таковые существуют) вершины u_i верно $M, u_{i+1} \models \varphi$ [существует хотя бы один сын u_{i+1} вершины u_i, и верно $M, u_{i+1} \models \varphi$]

• если θ есть $\forall\Box\varphi$ $[\exists\Box\varphi]$, то $M, u_i \models \theta \iff$ для каждого полного пути [существует полный путь] в графе с началом в вершине u_i в каждой его вершине u_j верно $M, u_j \models \varphi$

• если θ есть $\forall\Diamond\varphi$ $[\exists\Diamond\varphi]$, то $M, u_i \models \theta \iff$ для каждого полного пути [существует полный путь] в графе с началом в вершине u_i найдется вершина u_j этого пути, для которой верно $M, u_j \models \varphi$

• если θ есть $\forall(\varphi \mho \psi)$ $[\exists(\varphi \mho \psi)]$, то $M, u_i \models \theta \iff$ для каждого полного пути [существует полный путь] в графе с началом в вершине u_i найдется вершина u_j этого пути, для которой верно $M, u_j \models \psi$, а в каждой вершине этого пути u_k, из которой достижима u_j, отличной от u_j и достижимой из u_i верно $M, u_k \models \varphi$

Формула θ *истинна в модели* M, если она истинна в выделенной вершине u_0 этой модели. Формула θ *выполнима*, если она истинна в некоторой модели. Формула θ *общезначима*, если она истинна в каждой модели.

В дальнейшем $\top$ означает сокращение для формулы $(p \vee \neg p)$, а $\varphi \equiv \psi$ - для формулы $((\varphi \to \psi) \wedge (\psi \to \varphi))$.

Следующие формулы называются *аксиомами* (*схемами аксиом*) логики ветвящегося времени.

(Ax1) Корректная и полная система аксиом логики высказываний
(Ax2) $\exists\Diamond\varphi \equiv \exists(\top \mho \varphi)$
(Ax3) $\forall\Diamond\varphi \equiv \forall(\top \mho \varphi)$
(Ax4) $\forall\Box\varphi \equiv \neg\exists\Diamond\neg\varphi$
(Ax5) $\exists\Box\varphi \equiv \neg\forall\Diamond\neg\varphi$
(Ax6) $\exists\bigcirc(\varphi \vee \psi) \equiv (\exists\bigcirc\varphi \vee \exists\bigcirc\psi)$
(Ax7) $\forall\bigcirc\varphi \equiv \neg\exists\neg\varphi$
(Ax8) $\exists(\varphi \mho \psi) \equiv (\psi \vee (\varphi \wedge \exists\bigcirc\exists(\varphi \mho \psi)))$
(Ax9) $\forall(\varphi \mho \psi) \equiv (\psi \vee ((\varphi \wedge \forall\bigcirc\forall(\varphi \mho \psi)) \wedge \exists\bigcirc\top))$
(Ax10) $\forall\bigcirc\top$

Правила вывода в логике ветвящегося времени следующие:
(R1) $(\varphi \to \psi) \vdash (\exists\bigcirc\varphi \to \exists\bigcirc\psi)$
(R2) $(\chi \to (\neg\psi \wedge \exists\bigcirc\chi)) \vdash (\chi \to \neg\forall(\varphi \mho \psi))$
(R3) $(\chi \to (\neg\psi \wedge \forall\bigcirc(\chi \vee \neg\exists(\varphi \mho \psi)))) \vdash (\chi \to \neg\exists(\varphi \mho \psi))$
(R4) $\varphi, (\varphi \to \psi) \vdash \psi$

Формула φ называется *выводимой*, что обозначается $\vdash \varphi$, если существует такая конечная последовательность формул, заканчивающаяся формулой φ, в которой каждая формула является подстановкой в одну из аксиом или следует из предыдущих по одному из правил вывода.

Основные результаты.

Теорема. *Если формула* Θ *общезначима, то* $\vdash \Theta$.

Доказательство этой теоремы существенно опирается на доказательство полноты алгоритма распознавания выполнимости формул из [2], так как общезначимость формулы Θ равносильна невыполнимости формулы $\neg\Theta$. Вывод формулы Θ предъявляется.

Работа выполнена при финансовой поддержке РФФИ, проект 01-01-00930.

Литература

1. Emerson E. A., and Halpern J. Y., Decision Procedure and Expressiveness in the Temporal Logic of Branching Time, Journal of Computer and System Sciences, vol.30, no.1, pp.1-24, Feb.85.

2. Хелемендик Р. В. О корректности и полноте одного алгоритма распознавания выполнимости формул в логике ветвящегося времени.// Тезисы докладов XIII Международной конференции "Проблемы теоретической кибернетики" (Казань, 27-31 мая 2002 г., Часть II) - М.: Изд-во центра прикладных исследований при механико-математическом факультете МГУ, 2002, с. 183.

Задача о равномерном назначении минимальной стоимости

Н. Б. Чаплыгина (Ярославль)

В течение периода из m дней n работников: $b_1, b_2, \ldots, b_n$ выполняют s работ. В k-й день должны быть выполнены s_k работ ($\sum_{k=1}^{m} s_k = s$), задаваемых вектором $T^k = (t_1^k, \ldots, t_{s_k}^k)$, где $k \in \overline{1,m}$. Возможности выполнения работ заданы матрицей $R = (R^1, \ldots, R^m)$, где R^k – двумерная булева матрица размерности $n \times s_k$ с элементами

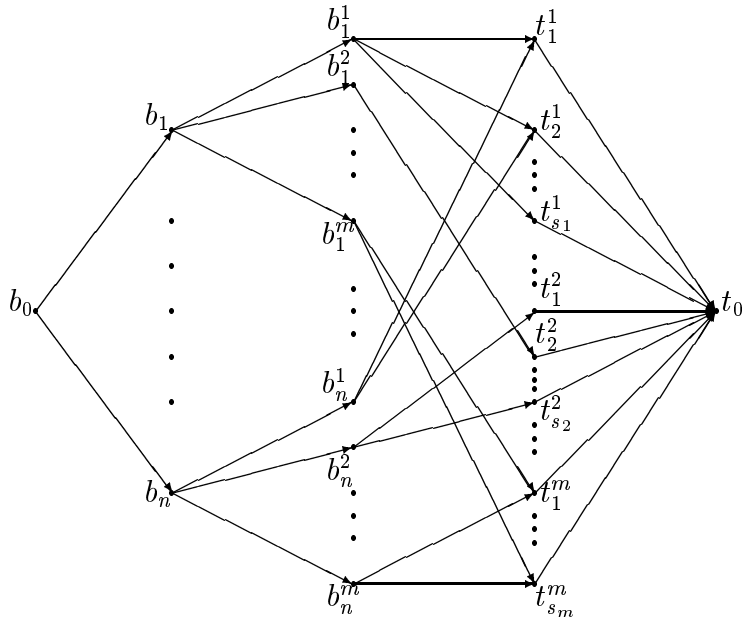
$$r_{ij}^k = \begin{cases} 1, \text{ если } i\text{-ый работник может в } k\text{-й день выполнить работу } t_j^k, \\ 0, \text{ если } i\text{-ый работник не имеет такой возможности.} \end{cases}$$

Обязательные назначения определим булевой матрицей A тех же размерностей, что и R. Матрица C задает стоимости выполнения работ каждым из рабочих. Допустимое назначение представим также булевой матрицей X, тогда его стоимость равна $\psi(X) = \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{s_k} c_{ij}^k x_{ij}^k$.

Допустимому решению X сопоставим итоговый вектор $X = (x_1, \ldots, x_n)$ (обозначим его тем же символом; из контекста обращения к X будет понятно, какой объект имеется в виду), где $x_i = \sum_{k=1}^{m} \sum_{j=1}^{s_k} x_{ij}^k$.

Необходимо среди всех назначений минимальной стоимости найти наиболее *равномерное,* т.е. минимизирующее функционал $L(X) = \sum_{i=1}^{n} (x_i - \overline{s})^2$, где $\overline{s} = \frac{1}{n} \sum_{i=1}^{m} s_i$.

Графически условие задачи можно представить в виде сети NA, которая имеет следующий вид:



В построенной по условиям задачи сети NA на дугах вида (b_0, b_i) пропускная способность не ограничивается, а на остальных дугах она равна 1. Стоимости дуг вида (b_i^k, t_j^k)

представлены в матрице C, остальные дуги имеют нулевую стоимость.Требуется найти поток величины s (т.е. максимальный), удовлетворяющий некоторым дополнительным условиям (обязательным назначениям матрицы A), минимизирующий функционал ψ и наиболее равномерно распределенный согласно критерия L.

Если X - допустимое решение минимальной стоимости, то в дополняющей поток X сети нет контуров отрицательной стоимости [1-2]. Наличие любого контура в дополняющей сети означает возможность переназначения работ по данному контуру. Если контур имеет положительную стоимость, то такое переназначение приведет к увеличению стоимости потока. Если контур имеет нулевую стоимость и проходит через вершины b_j, b_0, b_i именно в таком направлении и порядке и к тому же выполняется неравенство

$$x_i < x_j - 1, \tag{1}$$

то переназначение работ согласно данному контуру приведет к решению с меньшим значением функционала L, т.е. более равномерному. Такие контуры будем называть улучшающими. Утверждается, что если два решения максимальной величины и минимальной стоимости не имееют улучшающих контуров в дополняющих их сетях соответственно, то их итоговые векторы равны между собой с точностью до перестановки, и как следствие этого: если решение величины s минимальной стоимости не имеет улучшающих контуров, то оно является оптимальным по равномерности среди решений минимальной стоимости, т.е. решением поставленной задачи.

С учетом этого можно предложить алгоритм нахождения наиболее равномерного назначения минимальной стоимости:

1) построить максимальный поток;

2) пока есть контуры отрицательной стоимости, добавляя циркуляцию по такому контуру, уменьшать стоимость потока; если отрицательных контуров нет, то получен поток минимальной стоимости;

3) пока есть улучшающий контур: нулевой стоимости, количество назначений начальной рабочей вершины которого, увеличенное на 1, меньше количества назначений конечной рабочей вершины контура, т.е. выполнено условие (1), произвести переназначение по данному контуру; при этом стоимость потока не изменяется, а значение функционала L уменьшается на положительное целое число.

Поскольку $L(X) \geq 0$, то алгоритм закончит свою работу, полученный при этом поток будет соответствовать наиболее равномерному решению задачи A среди максимальных потоков минимальной стоимости.

Литература

1. Басакер Р., Т.Саати Т. Конечные графы и сети//М.,"Наука",1974.

2. Ху Т., Целочисленное программирование и потоки в сетях//М.,"Мир",1974.

3. Кропанов В. А., Рублев В. С., Равномерное назначение работ минимальной стоимости //Дискретная математика, 2001. Т.13, 4, С.144-156.

4. Кропанов В. А., Рублев В. С., Задача о равномерном назначении работ и ее обобщения //Моделирование и анализ информационных систем. Ярославль, 2000. Т.7, 2, С.3-12.

5. Рублев В. С., Чаплыгина Н.Б. Расширение задачи о назначениях //Моделирование и анализ информационных систем. Ярославль, 2002. Т.9, 2, С.3-11.

6. Рублев В.С. Задача о равномерном распределении работ //Ярославский госуниверситет. Ярославль, 1986.-Деп. ВИНИТИ 611-В87 26.01.87

Интерфейс пользователя динамической информационной модели DIM и навигатор объектов

А. Р. Юсупов (Ярославль)

Динамическая Информационная Модель (DIM), как и любая другая информационная система нуждается в удобном инструменте ввода информации и средстве навигации по имеющейся информации. Решать эту проблему созданием статичных форм неприемлемо, поскольку DIM является динамической системой, а в этом случае статичные запросы и формы имеют свойство устаревать, что и исключает их в большинстве случаев из применения в данной ситуации. Кроме этого сама структура данных DIM весьма сложна для подобного рода ее использования (о структуре данных DIM и об ее основных понятиях см. [1-2]).

Решать эту проблему возможно путем создания генератора входных форм основывающегося на схеме классов, в которой прописаны сами классы и связи между ними. Данная схема описывает в большей степени и поведение объектов этих классов, поскольку является шаблоном для объектов системы. Подобные попытки решения предпринимались ранее и одна из них (см. [3]) описывает навигатор с возможностью перемещаться по схеме классов DIM и возможностью создания объектов, редактирования их свойств, удаления и работой со связями с другими объектами. Казалось бы, все замечательно - навигатор с такими возможностями обеспечивает работу с данными системы DIM, но с другой стороны он не может гибко настраиваться и не обеспечивает работу с пользователями системы, что, в случае использования системы DIM для каких-либо целей очень важно.

Необходимо ввести кроме понятия системы и ее данных понятие *пользователь*. Именно пользователь работает с системой, и именно он вводит и получает данные системы. Кроме понятия *пользователь* необходимо определить также понятие *задача*. Задача - это то, с чем работает пользователь для ввода и получения информации в различных случаях. Задача связана со схемой классов, но охватывает ее не всю, а в общем случае лишь часть. К примеру, можно рассмотреть задачу по ведению справочника продукции и задачу, которая работает на основе этого справочника (например, складской учет). Функции у этих задач совершенно разные, несмотря на то, что некоторые классы, которые они используют, одинаковые. Пользователь же работает с системой на основе прав, то есть имеет права как на определенную задачу, так и на ее составные части.

Естественно, совершенно необходим механизм связи системы и пользователей. На эту роль весьма хорошо подходит так называемая связь *взаимодействия* (см. [4]), где одним из участников может быть пользователь, а другим система.

По этой технологии задача описывается взаимодействием, пользователю можно дать на нее права и тогда эта задача появится в его пользовательском меню и он сможет ее запускать и выполнять с помощью нее определенные действия в зависимости от прав (либо просматривать объекты, либо еще их и добавлять/редактировать и т.д.).

Рассмотрим проблему создания пользовательского интерфейса. Интерфейс должен быть общим для всех задач, автоматически должна создаваться форма для просмотра информации и формы для ввода/редактирования информации. Форма для просмотра информации должна содержать как иерархическое дерево объектов класса, если в данном классе присутствует вложенная иерархия/наследование, так и список объектов класса с их свойствами и свойствами, наследуемыми от объектов-родителей. Для каждого объекта необходимо предоставить возможность просмотра вложенных в него объектов с их свойствами (в том числе и наследуемыми).

В построении интерфейса существуют две крайности: во-первых, можно для каждого пользователя гибко настроить каждое взаимодействие, чтобы он мог видеть и делать только то, что ему положено, используя права, но пользователь теряет возможность сам настраивать приложение *под себя*, это делает только администратор системы, а во-вторых, можно определить задачу так сказать *в общем*, но тогда вообще теряется возможность каким-либо образом настроить приложение под пользователя.

Видимо наиболее оптимальным является вариант, обобщающий оба предыдущих: взаимодействие настраивается администратором системы под некоего *общего* пользователя, а пользователь может неким ограниченным образом модифицировать форму под себя (например, модифицировать размещение, увеличивать/уменьшать количество данных на форме, в пределах предусмотренным взаимодействием и правами пользователя).

При этом решении администратором вводится некий шаблон, который описывает задачу целиком и привязывается к взаимодействию, управляющему этой задачей. Модифицируя пользовательские права можно добиться управления пользователями: разрешить/запретить запуск взаимодействия (задачи), разрешить/запретить просмотр определенных частей данных задачи, разрешить/запретить добавление или модифицирование данных задачи. Кроме этого пользователь может настроить под себя все формы, указав нужно ли ему отображение тех или иных данных, в каком порядке ему нужно это представление и т.д.. Причем эта настройка сохраняется для каждого пользователя в самом взаимодействии.

Литература

1. Рублев В.С., Дерябин В.О., Лобачев Д.И., Юсупов А.Р. Базовые отношения объектов баз данных и гибкие таблицы, Моделирование и анализ информационных систем.// Ярославль, 2002. Т.9, N 2. с. 16-27.

2. Рублев В.С., Дерябин В.О., Иоссель М.А., Карповский А.В., Лобачев Д.И., Юсупов А.Р. Классы отношений объектов и взаимодействия объектов. Дискретные модели в теории управляющих систем. Тезисы докладов V научной конференции.// (Дубна, 26-29 мая 2003).

3. Юсупов А.Р. Генерация входных форм динамической информационной модели. Современные проблемы математики и информатики: Сборник научных трудов молодых ученых, аспирантов и студентов. Вып.5.// Яросл. гос. ун-т. Ярославль, 2002. С. 120-127.

4. Лобачев Д.И., Рублев В.С. Взаимодействия динамической информационной модели. Дискретные модели в теории управляющих систем. Тезисы докладов V научной конференции.// (Дубна, 26-29 мая 2003).

Об оценках для l-упаковок большого радиуса

М. С. Ярыкина (Москва)

Код C называется *l-упаковкой,* если каждый шар радиуса R содержит не более чем l наборов из кода C.

Если при передаче кодового слова в нем произошло не более R ошибок, мы сможем декодировать его списком из l элементов, содержащим кодовое слово.

Постановка задачи.

Для l-упаковки C найти зависимость между ее мощностью и радиусом R. Рассматривается случай $R = \tau n$, где $\tau > \frac{1}{4}$, причем ищутся оценки вида $\tau > f(m)$ или $m > f(\tau)$, при выполнении которых обязательно существует шар веса $l+1$ радиуса τ для кода C мощности m для любого n, т. е. не существует l-упаковки с таким соотношением параметров.

Случай $\tau < \frac{1}{4}$ подробно изучался, например в [1, 2], для случая $\tau > \frac{1}{4}$ автору известно только неравенство Плоткина $m > 1 + \frac{1}{4\tau - 1}$ для $l = 1$.

Полученный результат.

Не существует l-упаковок при следующих соотношениях параметров:

$$\tau > \frac{1}{2}\left(1 - 2\frac{\binom{\frac{m}{2}}{l+1}}{\binom{m}{l+1}}\right).$$

Или, при $\frac{1}{2} - \frac{1}{2^{l+1}} < \tau < \frac{1}{2}$, получаем

$$m > \frac{\frac{l(l+1)}{2}}{(2\tau-1)2^l + 1} + \frac{l(l+1)}{2} + Const \cdot 2^l\left(2\tau - 1 + \frac{1}{2^l}\right),$$

где $Const$ — некоторая константа, зависящая от l.

При конкретных малых l эти условия приобретают следующий вид. При $l = 1$ — это в точности неравенство Плоткина,

при $l = 2$

$$\tau > \frac{3}{8}\left(1 + \frac{1}{m-1}\right), \text{ или } m > 1 + \frac{3}{8\tau - 3};$$

при $l = 3$

$$\tau > \frac{1}{16}\left(7 + \frac{6m-21}{m^2-4m+3}\right) \text{ или } m > 2 + \frac{3}{16\tau - 7} + \sqrt{\left(\frac{3}{16\tau-7} + \frac{5}{6}\right)^2 + 2\frac{11}{36}};$$

при $l = 4$

$$\tau > \frac{1}{32}\left(15 + \frac{10m-45}{m^2-4m+3}\right) \text{ или } m > 2 + \frac{5}{32\tau - 15} + \sqrt{\left(\frac{5}{32\tau-15} + \frac{1}{2}\right)^2 + 2\frac{3}{4}};$$

Краткое доказательство.

Докажем, что при данных условиях существует шар радиуса $R = \tau n$ веса больше $l+1$. Для этого построим подкод C' мощности $(l+1)$ кода C и докажем, что весь C' попадает в шар требуемого радиуса.

Строим подкод C' следующим образом. Выберем в коде C любые $(l+1)$ двоичных набора ($l+1 \le \frac{m}{2}$), совпадающие не менее, чем в t компонентах, где

$$t = \frac{n}{\binom{m}{l+1}} \cdot \left(\binom{\lfloor m/2 \rfloor}{l+1} + \binom{\lceil m/2 \rceil}{l+1} \right) \approx \frac{2n\binom{m/2}{l+1}}{\binom{m}{l+1}}.$$

Получили подкод $C' \subseteq V^{n-t}$ мощности $(l+1)$. Средний вес шара радиуса R кода C' равен:

$$P_R(C') = \frac{1}{2^{n-t}}(l+1)\sum_{i=0}^{R}\binom{n-t}{i} = (l+1) - \frac{l+1}{2^{n-t}}\sum_{i=0}^{n-t-R-1}\binom{n-t}{i}.$$

Чтобы накрыть весь C' одним шаром, радиус шара должен удовлетворять следующему условию:

$$R > \frac{1}{2}(n-t), \quad \text{то есть} \quad \tau > \frac{1}{2}\left(1 - 2\frac{\binom{m/2}{l+1}}{\binom{m}{l+1}}\right).$$

При $l = 1$ это в точности соответствует неравенству Плоткина, а при $l > 1$

$$\frac{\binom{m/2}{l+1}}{\binom{m}{l+1}} = \frac{1}{2^{l+1}}\frac{1\left(1-\frac{2}{m}\right)\left(1-\frac{4}{m}\right)\ldots\left(1-\frac{2l}{m}\right)}{1\left(1-\frac{1}{m}\right)\left(1-\frac{2}{m}\right)\ldots\left(1-\frac{l}{m}\right)} = \frac{1}{2^{l+1}}\left(1 - \frac{\frac{l(l+1)}{2}}{m - \frac{l(l+1)}{2} + \frac{\text{const}}{m^2}}\right).$$

Обозначим $C = \frac{l(l+1)}{2}$. Тогда получим

$$R > \frac{1}{2}(n-t) = \frac{1}{2}\left(n - \frac{2n}{2^{l+1}}\left(1 - \frac{C}{m-C} + \frac{\text{const}}{m^2}\right)\right) = \frac{1}{2}n\left(1 - \frac{1}{2^l} + \frac{1}{2^l}\frac{C}{m-C} + \frac{\text{const}}{m^2}\right).$$

Применяя выражения для τ, с помощью преобразований получим

$$\begin{aligned} 2\tau - \left(1 - \frac{1}{2^l}\right) &> \frac{1}{2^l}\left(\frac{C}{m-C} + \frac{\text{const}}{m^2}\right), \\ \frac{1}{(2\tau-1)2^l + 1} &< \frac{m-C}{C + \text{const}\cdot 2^l\left(2\tau - 1 + \frac{1}{2^l}\right)}, \\ m &> \frac{\frac{l(l+1)}{2}}{(2\tau-1)2^l+1} + \frac{l(l+1)}{2} + \text{const}\cdot 2^l\left(2\tau - 1 + \frac{1}{2^l}\right). \end{aligned}$$

Автор выражает благодарность научному руководителю Ю. В. Таранникову за постановку задачи, внимание к работе и ценные советы. Работа поддержана грантами РФФИ 02-01-00985 и Университеты России УР.04.03.007.

Литература

1. Blinovsky V. Bounds for Multiple Packing in q-ary Hamming Space.// Proceedings of ISIT 2002, Lausanne, Switzerland, June 30 – July 5, 2002, P. 401.

2. Ashikhmin A., Barg A., Litsyn S. A new upper bound on codes decodable into size-2 lists.// Numbers, Informatiosn and Complexity, pp. 239–244. Boston: Kluwer publishers, 2000.

LINSPACE-конструктивность алгебраических чисел

С. В. Яхонтов (Санкт-Петербург)

Введение.

Понятие конструктивного действительного числа является вычислительным уточнением понятия абстрактного действительного числа. Конструктивные действительные числа позволяют находить сколь угодно точные рациональные аппроксимации действительных чисел. Вычислительная сложность конструктивных действительных чисел подробно изучалась в ряде работ, среди которых можно отметить, например, [4, 5]. Общим свойством этих работ является изучение временной сложности проблем численного анализа, при этом основное внимание уделяется вычислимости за полиномиальное время.

Основные определения.

При изучении вычислительной сложности конструктивных действительных чисел и функций удобно в качестве множества аппроксимирующих значений использовать множество D двоично-рациональных чисел. Под точностью $prec(s)$ представления

$$s = \pm s_n s_{n-1} \dots s_0 . t_1 t_2 \dots t_m$$

двоично - рационального числа $d = \pm \sum_{i=0}^{n} s_i 2^i \pm \sum_{j=1}^{m} t_j 2^{-j}$ понимается число битов справа от двоичной точки.

Функция $\phi : N \to D$ двоично-рационально сходится к действительному числу x, если $\forall n \in N : prec(\phi(n)) = n$ и $|\phi(n) - x| \leq 2^{-n}$. Множество всех функций, двоично-рационально сходящихся к действительному числу x, обозначается CF_x. Действительное число x называется CF-вычислимым ([4]), если CF_x содержит вычислимую функцию ϕ. В качестве длины входных данных при оценке вычислительной сложности функций $\phi \in CF_x$ берется n.

Класс алгоритмов, для которых объем промежуточных вычислений ограничен линейной функцией от длины входных данных, называется *LINSPACE* ([3]).

Определение. CF-вычислимое действительное число x будем называть LINSPACE конструктивным действительным числом, если существует функция $\phi \in CF_x$ класса *LINSPACE*.

Множество LINSPACE конструктивных действительных чисел будем обозначать $LINSPACE_{CF}$.

***LINSPACE*-конструктивные алгебраические числа.**

Для построения конструктивных алгебраических чисел будем использовать стандартное представление вещественных корней полинома $P(x) \in Z[x]$ списком отделяющих интервалов. Данное представление позволяет вычислять рациональные приближения к алгебраическим числам с любой заданной точностью. Для решения задачи нахождения списка отделяющих интервалов воспользуемся алгоритмом Штурма ([1]).

Алгоритм *SH* (алгоритм Штурма).

Вход: список $(p_0, \dots, p_n)$ коэффициентов полинома $P(x) = \sum_{i=0}^{n} p_i x^i$

Выход: список $I = ((a_1, b_1], \dots (a_\nu, b_\nu])$ отделяющих интервалов

Реализация:

1. верхняя граница корней $u := 2^c > 1 + \frac{\max\{|p_1|,\dots,|p_n|\}}{|p_0|}$, список $I := ((-u, u])$

2. вычислить коэффициенты полиномов субрезультантной последовательности Штурма $(P_1 = P, P_2 = P', P_3, \ldots, P_k)$ ([2])

3. выполнить шаги 3.a - 3.c для всякого интервала $(a, b]$ в списке I:

   (a) определить бисекцию интервала $(a, b]$ на подинтервалы $(a, m]$ и $(m, b]$

   (b) вычислить число вещественных корней $\rho_{(a,m]}$ и $\rho_{(m,b]}$ на $(a, m]$ и $(m, b]$

   (c) для каждого из подинтервалов: если $\rho = 0$, то удалить подинтервал; если $\rho = 1$, то дать подинтервал на выход; если $\rho > 1$, то оставить подинтервал в списке I

Теорема 1. *Пусть* $P(x) = \sum_{i=0}^{n} p_i x^i$ *- примитивный полином положительной степени с целыми коэффициентами, свободный от квадратов. Тогда емкостная сложность алгоритма SH, отображающего список коэффициентов полинома* $P(x)$ *в список отделяющих интервалов вещественных корней этого полинома, ограничена функцией*

$$C(n^4 + n^3 \cdot l(\max_{i=0}^{n} |p_i|)).$$

Алгоритм *SH* позволяет по произвольнолму полиному $P \in Z[x]$ построить набор CF - вычислимых действительных чисел, являющихся конструктивными аналогами корней полинома $P(x)$. Для вычисления рациональных приближений к корню α можно взять итеративный алгоритм, на каждом шаге которого интервал, содержащий α, делится на два подинтервала, затем выбирается тот из подинтервалов, на котором полином меняет знак, и выдается двоично-рациональная концевая точка выбранного подинтервала.

Теорема 2. *Пусть* $P(x) \in Z[x]$ *- примитивный полином положительной степени, свободный от квадратов,* $\{\alpha_i\}$ *- набор вещественных корней полинома* $P(x)$*. Тогда все числа набора* $\{\alpha_i\}$ *принадлежат классу* $LINSPACE_{CF}$*.*

Литература

1. Б.Бухбергер, Дж.Коллинз, Р.Лоос, Компьютерная алгебра. Символьные и алгебраические вычисления, Мир, 1986.

2. Д.Кнут, Искусство программирования. Получисленные алгоритмы, изд.дом "Вильямс", 2001.

3. D.Du, K.Ko, Theory of Computational Complexity, John Wiley & Sons, 2000.

4. K.Ko, Complexity Theory of Real Functions, Birkhauser, 1991.

5. K.Ko, Polynomial-time computability in analysis, in "Handbook of Recursive Mathematics," Volume 2, Recursive Algebra, Analysis and Combinatorics, 1998, pp.1271-1317.