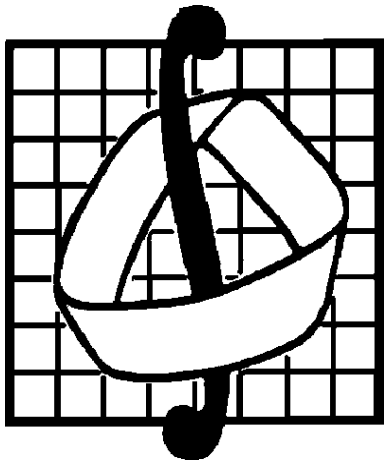
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА



Механико-математический факультет

МАТЕРИАЛЫ
IX Международной конференции "Интеллектуальные системы и компьютерные науки"

(23-27 октября 2006 г.)

ТОМ 1
часть 2

Издательство механико-математического факультета МГУ
2006

УДК 519.95, 519.14, 519.1, 519.6

Издание осуществлено при поддержке
Российского фонда фундаментальных исследований
по проекту № 06-01-10-114

Материалы IX Международной конференции "Интеллектуальные системы и компьютерные науки"(23-27 октября 2006 г.), том 1, часть 2. - М.: Изд-во механико-математического факультета МГУ, 2006.

Сборник содержит работы участников IX Международной конференции "Интеллектуальные системы и компьютерные науки", проходившей на механико-математическом факультете МГУ им. М. В. Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114). Сборник адресован научным сотрудникам, преподавателям, аспирантам и студентам, работающих и интересующихся тематикой математических проблем теории интеллектуальных систем и их приложений.

Научное издание

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ"(23-27 октября 2006 г.) Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В.

В составлении и редактировании сборника принимали участие: Строгалов А. С., Носов В. А., Уварова Т. Д., Холоденко А. В., Галатенко А. А.

Ответственный за выпуск Строгалов А. С.

Вероятностный анализ различных шельфовых алгоритмов упаковки прямоугольников в полосу [1]

Кузюрин Н. Р., Поспелов А. И., E-mail: {nnkuz,ap}@ispras.ru
ИСП РАН, 109004, Москва, Б.Коммунистическая, 25

В задаче упаковки в полосу (strip packing) цель состоит в упаковке множества прямоугольников в вертикальную полосу единичной ширины так, что стороны прямоугольников должны быть параллельны сторонам полосы (вращения запрещены). При анализе по худшему случаю обычно минимизируют необходимую для упаковки высоту полосы, а при анализе в среднем – математическое ожидание незаполненной площади (от нижней границы прямоугольников в упаковке до верхней) [8,3]. Эта задача возникает во многих контекстах и имеет много приложений, в частности, при разработке СБИС, построении оптимальных расписаний для кластеров и т.д. Ее частными случаями являются задача упаковки в контейнеры (bin packing) и задача об m-процессорном расписании.

При анализе on-line версии задачи упаковки в полосу в [1] были введены так называемые шельфовые алгоритмы. Упаковка состоит из серии слочв (шельфов). Высоты слочв выбираются из множества $\{r_n\}$, где $r_n = (1-\delta)^n$, δ – некоторый параметр, $0 < \delta < 1$. Прямоугольники упаковываются в минимальные по высоте шельфы, в которые они входят, т.е. если высота прямоугольника h_i, упаковываем его в шельфы высоты r_n, такой что $r_{n+1} < h_i \leq r_n$. Упаковка в слои заданной ширины и порождение новых шельфов осуществляется некоторой одномерной эвристикой упаковки в контейнеры.

Мы будем называть алгоритмом **A(E)** шельфовый алгоритм упаковки, который на втором этапе использует некоторую произвольную эвристику **E**. Цель работы состоит в описании общего метода вероятностного анализа шельфовых алгоритмов **A(E)**, позволяющего для многих шельфовых алгоритмов оценивать ожидаемую незаполненную площадь, используя соответствующие результаты для одномерной эвристики **E**.

Мы будем рассматривать вероятностное распределение $\mathrm{U}([0,1])$ — равномерное распределение на отрезке $[0,1]$. Всюду в дальнейшем будем считать, что для каждого прямоугольника высота h_i и ширина w_i имеют равномерное распределение на отрезке $[0,1]$. Будем предполагать, что все случайные величины w_i, h_i — независимы в совокупности. В дальнейшем будем обозначать через Σ математическое ожидание площади, не заполненной прямоугольниками, между основанием полосы и верхней границей самого верхнего шельфа. Будем предполагать, что число прямоугольников N – бесконечно большая величина ($N \to \infty$). Отметим, что вероятностному анализу различных эвристик одно и двумерной упаковки посвящено много работ [2,3,4,5,6,7].

Пусть для одномерной эвристики **E** выполнено соотношение

$$\mathbb{E}\left(L - \sum_{i=1}^{N} w_i\right) = O\left(f(N)\right),$$

где L - число шельфов, в которые эвристика **E** упаковывает набор отрезков $\{w_i\}$.

Теорема 1. *Пусть для* **E** $f(N) = N^{\alpha}\log^{\beta} N$, *где* $0 < \alpha < 1$, $\beta \geq 0$. *Тогда для алгоритма* **A(E)** *справедлива следующая оценка:*

$$\Sigma = O\left(N^{\alpha}\left(\log^{\beta} N/\delta^{1-\alpha} + \delta N^{1-\alpha}\right)\right).$$

Выбирая $\delta = N^{(\alpha-1)/(2-\alpha)}\log^{(\beta/(2-\alpha))} N$, *получаем оценку* $\Sigma = O\left(N^{1/(2-\alpha)}\log^{(\beta/(2-\alpha))} N\right)$.

Отметим, в частности, что для трех важных эвристик упаковки в контейнеры известно: для FF $f(N) = N^{2/3}$, для BF $f(N) = N^{1/2}(\log N)^{3/4}$, и для наилучшей онлайновой эвристики (best on-line) BO $f(N) = (N\log N)^{1/2}$ [5]. Поэтому из теоремы сразу вытекает, что для шельфового алгоритма **A(FF)** $\Sigma = O(N^{3/4})$, для **A(BF)** $\Sigma = O(N^{2/3}(\log N)^{1/2})$ [9], для **A(BO)** $\Sigma = O(N^{2/3}(\log N)^{1/3})$. Первые два соотношения дают ответ на вопрос из [6].

Основные моменты доказательства теоремы 1. Пусть зафиксированы N — число прямоугольников, и $\{r_n\}_{n=0}^{\infty}$ – высоты шельфов. Каждый i-й прямоугольник определчн своей высотой h_i и шириной w_i, которые являются независимыми в совокупности случайными величинами. Для краткости будем обозначать через **h** набор $(h_1, \ldots h_N)$, а через **w** – набор $(w_1, \ldots w_N)$. Пусть N_n обозначает число прямоугольников, попавших в шельфы высоты r_n, т.е.

$$N_n = \#\left\{h_i | r_{n+1} < h_i \leq r_n\right\}.$$

Заметим, что случайная величина N_n не зависит от **w**, а зависит только от **h**.

[1] Работа выполнена при поддержке РФФИ, проект 05-01-00798.

Пусть $\{w_i^n, h_i^n\}_{i=1}^{N_n}$ — набор прямоугольников, упакованных в шельфы высоты r_n. Так как h_i имеет равномерное распределение то случайная величина N_n имеет биномиальное распределение, такое что $P\{N_n = k\} = \binom{N}{k} p^k (1-p)^{N-k}$, где $p = r_n - r_{n+1}$, $0 \le k \le N$. Пусть S_n — число шельфов высоты r_n, образовавшихся в процессе упаковки.

Используя для условного математического ожидания $\mathbb{E}(X|Y)$ равенство $\mathbb{E}\, X = \mathbb{E}(\mathbb{E}(X|Y))$, оценим ожидаемую пустую площадь:

$$\mathbb{E}\left[\sum_{n=0}^{\infty} r_n S_n - \sum_{i=1}^{N} w_i h_i\right] = \mathbb{E}\sum_{n=0}^{\infty} r_n S_n - \frac{N}{4} = \mathbb{E}\sum_{n=0}^{\infty} r_n\, \mathbb{E}\,(S_n \,|\, \mathbf{h}) - \frac{N}{4}.$$

Учитывая предположение теоремы об используемой эвристике и оценке для нее математического ожидания незаполненной площади, получаем:

$$\mathbb{E}\left((S_n - \sum_{i=1}^{N_n} w_i^n)\,\middle|\,\mathbf{h}\right) = O\left(N_n^{\alpha} \log^{\beta} N_n\right).$$

Откуда, ввиду независимости w_i^n от $\mathbf{h}$, получаем оценку для числа шельфов

$$\mathbb{E}\,(S_n|\mathbf{h}) = \mathbb{E}\left(\sum_{i=1}^{N_n} w_i^n \,\middle|\, \mathbf{h}\right) + O\left(N_n^{\alpha} \log^{\beta} N_n\right) = \frac{N_n}{2} + O\left(N_n^{\alpha} \log^{\beta} N_n\right).$$

Используя это соотношение, получим, что математическое ожидание пустой площади равно

$$\mathbb{E}\sum_{n=0}^{\infty} r_n\, \mathbb{E}\,(S_n \,|\, \mathbf{h}) - \frac{N}{4} = \mathbb{E}\sum_{n=0}^{\infty} r_n \left[\frac{N_n}{2} + O\left(N_n^{\alpha} \log^{\beta} N_n\right)\right] - \frac{N}{4} =$$

$$\sum_{n=0}^{\infty}\left[r_n \frac{N(r_n - r_{n+1})}{2} + \mathbb{E}\, r_n O\left(N_n^{\alpha} \log^{\beta} N_n\right)\right] - \frac{N}{4} = \Sigma.$$

В последнем равенстве мы использовали очевидное соотношение: $\mathbb{E}\, N_n = N(r_n - r_{n+1})$. Учитывая, что $r_n = (1-\delta)^n$, $0 < \delta < 1$, получаем для некоторой константы $c > 0$:

$$\Sigma \le c \log^{\beta} N \sum_{n=0}^{\infty} (1-\delta)^n\, \mathbb{E}\, N_n^{\alpha} + \frac{N}{2(2-\delta)} - \frac{N}{4}. \tag{1}$$

Для получения верхней оценки воспользуемся неравенством Йенсена $\mathbb{E}\, N_n^{\alpha} \le (\mathbb{E}\, N_n)^{\alpha}$. Поскольку $p = r_n - r_{n+1} = (1-\delta)^n \delta$ и $\mathbb{E}\, N_n = Np$, то $\mathbb{E}(N_n)^{\alpha} \le (Np)^{\alpha}$.

Тогда оценим сумму в (1), обозначив ее через Σ_1.

$$\Sigma_1 \le \sum_{n=0}^{\infty} (1-\delta)^n\, \mathbb{E}\, N^{\alpha} \le \sum_{n=0}^{\infty} (1-\delta)^n (N(1-\delta)^n \delta)^{\alpha},$$

$$\Sigma_1 \le \sum_{n=0}^{\infty} (1-\delta)^n (N(1-\delta)^n \delta)^{\alpha} \le \sum_{n=0}^{\infty} (1-\delta)^n (N(1-\delta)^n \delta)^{\alpha} = \frac{(N\delta)^{\alpha}}{1 - (1-\delta)^{\alpha+1}}.$$

Таким образом, получаем оценку для Σ.

$$\Sigma \le c \log^{\beta}(N) \cdot (N\delta)^{\alpha} / 1 - (1-\delta)^{\alpha+1} + N/(2(2-\delta)) - N/4 \le$$

$$N^{\alpha}\left(\log^{\beta}(N) c \delta^{\alpha}(1 - (1-\delta)^{\alpha+1}) + \delta N^{1-\alpha}/4\right) \le N^{\alpha}\left(c \log^{\beta}(N)/(\delta^{1-\alpha}) + \delta N^{1-\alpha}/4\right).$$

Выбирая $\delta = N^{(\alpha-1)/(2-\alpha)} \log^{(\beta/(2-\alpha))} N$, получаем оценку $\Sigma = O\left(N^{1/(2-\alpha)} \log^{(\beta/(2-\alpha))}\right)$.

Список литературы

1. B.S. Baker, J.S. Schwartz, Shelf algorithms for two dimensional packing problems. SIAM J. Computing, (1983) **12**, 508–525

2. E.G. Coffman, Jr., C. Courcoubetis, M.R. Garey, D.S. Johnson, P.W. Shor, R.R. Weber, M. Yannakakis, Perfect packing theorems and the average-case bahavior of optimal and online bin packing, SIAM Review, (2002) **44** (1), 95–108.

3. R.M. Karp, M. Luby, A. Marchetti-Spaccamela, A probabilistic analysis of multidimensional bin packing problems, In: Proc. Annu. ACM Symp. on Theorty of Computing, 1984, pp. 289–298.

4. P. W. Shor, The average-case analysis of some on-line algorithms for bin packing. Combinatorica (1986) **6**, 179-200.

5. P. W. Shor, How to pack better than Best Fit: Tight bounds for average-case on-line bin packing, Proc. 32nd Annual Symposium on Foundations of Computer Science, (1991) pp. 752-759.

6. E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and G. S. Lueker. Probabilistic analysis of packing and related partitioning problems, Statistical Science (1993) **8**, 40-47.

7. E. G. Coffman, Jr., P. W. Shor. Packings in two dimensions: Asymptotic average-case analysis of algorithms, Algorithmica (1993) **9**, 253-277.

8. J. Csirik, G.J. Woeginger, Shelf Algorithms for On-Line Strip Packing. Inf. Process. Lett. (1997), **63**(4), 171-175.

9. Кузюрин Н.Н., Поспелов А.И., Вероятностный анализ шельфовых алгоритмов упаковки прямоугольников в полосу, Дискретная математика, 2006, т. 18, N 1, с. 76–90.

О сложности поиска идентичных объектов для случайных баз данных

Кучеренко Н. С., E-mail: kucherenko_n@tochka.ru

Московский Государственный Университет им. М.В. Ломоносова

Теория хранения и поиска информации является важным разделом теории интеллектуальных систем. Одним из ключевых объектов этой теории является информационный граф (ИГ) [1] — управляющая система, которая позволяет рассматривать имеющиеся модели данных и задачи, связанные с ними, с более общих позиций.

В данной работе в рамках информационно-графовой модели данных рассматривается задача поиска идентичных объектов (ЗПИО) на отрезке $(0,1)$. Формально ЗПИО на отрезке $(0,1)$ — это тройка $I = ((0,1), V, \rho_=)$, где V — конечное подмножество отрезка $(0,1)$, которое называется библиотекой, $\rho_=$ — отношение равенства [1]. С помощью информационных графов моделируются алгоритмы поиска в библиотеке, использующие только операции сравнения. В предположении, что на множестве запросов задано вероятностное пространство, для информационного графа U вводится понятие сложности $T(U)$. Сложность $T(U)$ — это математическое ожидание сложности ИГ на запросе [1]. Сложностью задачи $T(I)$ определяется как инфимум сложности всех ИГ, которые решают задачу I. ИГ, на котором достигается инфимум, называется *оптимальным.*

Автором был исследован вопрос существования оптимальных ИГ и получен следующий результат.

Теорема 1 *Для любой задачи $I = ((0,1), V, \rho_=)$ существует оптимальный ИГ. Его структура имеет вид дерева, с n не листовыми вершинами.*

Этот факт позволяет применить технику, развитую в работе [2], для построения оптимального графа за полиномиальное время от мощности библиотеки V. Предположим, что функция $F(x)$ распределения запроса — произвольная интегрируемая по Риману функция. Рассмотрим класс задач $I_n^F = ((0,1), V_n, \rho_=)$, где $V_n = \{\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}\}$.

Теорема 2 *Для любой интегрируемой по Риману функции F верно $T\left(I_n^F\right) \sim \log_2 n \quad (n \to \infty)$*

Рассмотрим случай, когда библиотека $V_n = (y_1, \ldots, y_n)$ является случайным вектором, компоненты которого независимы и равномерно распределены. Обозначим через $I_n = ((0,1), V_n, \rho_=)$ ЗПИО, в которой запросы также распределены равномерно. Нетрудно показать, что сложность задачи $T(I_n)$ — случайная величина. Математическое ожидание этой случайной величины обозначим через $\mathbf{M}\,T(I_n)$

Теорема 3 $\mathbf{M}\,T(I_n) = \log_2(n+1) + (\gamma_{n+1} - 1)/ln2$, *где* $\gamma_n = \sum_{i=1}^{n} \frac{i}{n} - \ln(n)$ *и последовательность* γ_n *сходится к постоянной Эйлера* $\gamma = 0,577\ldots$

Автор выражает благодарность своему научному руководителю профессору Гасанову Эльяру Эльдаровичу за постановку задачи, внимание к работе, ценные советы и обсуждения.

Список литературы

1. Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации — М.: Физматлит, 2002.

2. Knuth D. E. Optimum binary search trees // Acta Informatica — 1971. — Т. 1, — No 1, — С. 14-25.

О вероятностной мере, наименее уклоняющейся от симметричной части неточности

Лепский А. Е.,
докторант Таганрогского государственного радиотехнического университета,
347928, Таганрог, пер. Некрасовский, 44, ТРТУ, Лаборатория МПИИ (Д-517),
e-mail: lepskiy@mail.ru

1. Введение

Пусть X – некоторое конечное множество, $\mathbf{A}_X$ – множество всех подмножеств из X, $\mathcal{P}$ – некоторое непустое семейство вероятностных мер на $\mathbf{A}_X$. Тогда, следуя [1], для произвольной функции $f: X \to R$ можно по семейству $\mathcal{P}$ оценить снизу математическое ожидание – получим функционал $\underline{E}[f] = \inf\left\{\sum_{x\in X} f(x)P(x):\ P \in \mathcal{P}\right\}$, называемый нижним предвидением функции f. В частности, если $f(x) = \chi_A(x)$ – характеристическая функция множества $A \in \mathbf{A}_X$, то получим $\underline{E}[\chi_A] = \inf\{P(A): P \in \mathcal{P}\} = g(A)$ – нижнюю оценку вероятности события A относительно семейства $\mathcal{P}$. Функцию множеств g называют нижней вероятностью. Аналогично можно определить верхнее предвидение $\overline{E}[f] = \sup\left\{\sum_{x\in X} f(x)P(x): P \in \mathcal{P}\right\}$ и верхнюю оценку $\overline{E}[\chi_A] = \sup\{P(A): P \in \mathcal{P}\} = \overline{g}(A)$ вероятности события A относительно семейства $\mathcal{P}$. Тогда пара $(g(A), \overline{g}(A))$ будет определять неточность задания вероятности события A. В работе рассматривается задача нахождения вероятности, наиболее уклоняющейся от «границ» неточности $(g(A), \overline{g}(A))$, $A \in \mathbf{A}_X$. В определенном смысле эта вероятностная мера P будет оптимальной мерой, определяемой неточностью $(g, \overline{g})$.

2. Основные определения, обозначения и постановка задачи

Через $M_0(X)$ будем обозначать множество монотонных нормированных мер на $\mathbf{A}_X$, т.е. $g \in M_0(X)$, если:

1) $g(\varnothing) = 0$; $g(X) = 1$;

2) из $A, B \in \mathbf{A}_X$, $A \subseteq B$ следует, что $g(A) \le g(B)$.

Меру $\overline{g}$, определяемую равенством $\overline{g}(A) = 1 - g(\overline{A})$, $\forall A \in \mathbf{A}_X$, называют мерой, сопряженной к мере g. Нетрудно видеть, что $\overline{g} \in M_0(X) \Leftrightarrow g \in M_0(X)$ и $\overline{\overline{g}} = g$.

Пусть $P(X)$ - множество всех вероятностных мер на $\mathbf{A}_X$, а $M_{low}(X)$ - множество так называемых *нижних вероятностей* на $\mathbf{A}_X$, т.е. таких мер $g \in M_0(X)$, для которых существует $P \in P(X)$ такая, что $g(A) \le P(A)$ для всех $A \in \mathbf{A}_X$. Если $g \in M_{low}(X)$, то меру $\overline{g}$ называют *верхней вероятностью*.

Важным классом нижних вероятностей является класс так называемых примитивных мер, т.е. мер вида $\eta_{\langle B\rangle}(A) = \begin{cases} 1, B \subseteq A, \\ 0, B \text{ Ш} A \end{cases}$, $A, B \in \mathbf{A}_X$, $(B \ne \varnothing)$ [2] (подобная конструкция рассматривается и в кооперативной теории игр [3]). Заметим, что, используя так называемое преобразование Мёбиуса [4] $m_g(D) = \sum_{A: A\subseteq D} (-1)^{|D\setminus A|} g(A)$, $D \in \mathbf{A}_X$, любую функцию множеств можно разложить по монотонным примитивным мерам: $g = \sum_D m_g(D)\eta_{\langle D\rangle}$. В данной работе нам понадобятся примитивные меры $\eta_{\langle B\rangle}$ в случае, когда $|B| = 1$. Для таких примитивных мер будем использовать обозначения $\eta_i = \eta_{\langle\{x_i\}\rangle}$, $x_i \in X$.

Если для нормированной функции множеств g ($g(\varnothing) = 0$, $g(X) = 1$) ее преобразование Мебиуса m_g является неотрицательной функцией множеств, т.е. $m_g(D) \ge 0$ для всех $D \in \mathbf{A}_X$, то такую меру (которая будет нижней вероятностью) называют *мерой (или функцией) доверия* (belief measure) [5]. Множество всех мер доверия на $\mathbf{A}_X$ будем обозначать через $Bel(X)$.

Нижнюю вероятность $g \in M_{low}(X)$ можно рассматривать как нижнюю оценку вероятности P, а двойственную ей меру $\overline{g}$ можно считать верхней оценкой вероятности P. Таким образом, любая нижняя вероятность $g \in M_{low}(X)$ определяет некоторый «отрезок», содержащий вероятность P: $g \le P \le \overline{g}$. Чем меньше разность $\overline{g} - g$, тем точнее мера g определяет некоторую вероятность. Поэтому «расстояние» между мерами $\overline{g}$ и g характеризует неточность определения вероятности, задаваемой мерой g.

Таким образом, нижняя вероятность g порождает на $M_0(X)$ функцию множеств $\nu_g = \overline{g} - g$. Для любого множества $A \in \mathbf{A}_X$ функция множеств $\nu_g(A)$ определяет величину (меру) неопределенности задания вероятности события A. Очевидно, что $\nu_g(A) = \nu_g(\overline{A})$ для всех $A \in \mathbf{A}_X$, $\nu_g(\varnothing) = \nu_g(X) = 0$ и $\nu_P \equiv 0$ для вероятностной меры P.

С другой стороны, мера $\rho_g = 0.5(g + \overline{g})$ характеризует «середину» неточности, определяемой функцией множеств ν_g. Очевидно, что $\rho_g \in M_0(X) \Leftrightarrow g \in M_0(X)$ и $g \le \rho_g \le \overline{g}$. Кроме того, пара функций множеств ν_g и ρ_g однозначно определяет саму меру g, поскольку $g = \rho_g - 0.5\nu_g$. В теории неточных вероятностей мера ρ_g называется *симметричной частью* пары $(g, \overline{g})$ (symmetric part) [6] или *центральной компонентой* (central component) [7] неопределенности $(g, \overline{g})$.

Нетрудно видеть, что мера ρ_g удовлетворяет следующим свойствам:

1) $\rho_g(A) + \rho_g(\overline{A}) = 1$, $A \in \mathbf{A}_X$, (свойство симметричности меры ρ_g). Это свойство можно трактовать как $\rho_g = \overline{\rho_g}$, т.е. мера ρ_g является самосопряженной;

2) если ρ_g - нижняя вероятность или верхняя вероятность, то ρ_g - вероятностная мера;

3) $\sum_{A \subseteq X} \rho_g(A) = 2^{|X|-1}$.

Тогда задачу нахождения вероятностной меры, наиболее уклоняющейся от «границ» $g(A)$ и $\overline{g}(A)$, $A \in \mathbf{A}_X$, неточности $(g, \overline{g})$ в классе нижних вероятностей можно сформулировать следующим образом. Требуется найти такую вероятностную меру P, чтобы величина $\|P - \rho_g\|$ была минимальной. Здесь $\|\cdot\|$ - некоторая норма в метрическом пространстве $R^{2^{|X|}-2}$. В работе поставленная задача решена относительно евклидовой нормы $\| \ \|_2$, т.е. найдена вероятностная мера P, минимизирующая функционал среднеквадратичного отклонения от ρ_g.

3. Основной результат

Пусть $|X| = n$. Тогда справедлива следующая теорема.

Теорема. *Для любой меры доверия* $g \in Bel(X)$ *существует единственная вероятностная мера* P *такая, что норма* $\|P - \rho_g\|_2$ *- минимальна, причем* $P = \sum_{k=1}^{n} \alpha_k \eta_k$, *где*

$$\alpha_k = \frac{1}{n} + \frac{1}{2^{n-2}} \sum_{A \subseteq X} \rho_g(A) \left(\eta_k(A) - \frac{|A|}{n} \right), \qquad (1)$$

$k = 1, \ldots, n$.

Следствие. *Для коэффициентов* α_k *оптимальной вероятностной меры* P *справедливы следующие представления:*

а) $\alpha_k = \frac{1}{n} \sum_{B \subseteq X \setminus \{x_k\}} \left[\left(1 - \frac{|B|}{2^{|B|-1}} \right) m_g(B) + \left(1 + \frac{n - |B| - 1}{2^{|B|}} \right) m_g(B \cup \{x_k\}) \right]$, $k = 1, \ldots, n$,

где m_g *- преобразование Мебиуса меры* g *;*

б) $\alpha_k = \frac{1}{n} + \frac{1}{n2^{n-2}} \sum_{A \subseteq X \setminus \{x_k\}} \left[(n-1-|A|) g(A \cup \{x_k\}) - |A| g(A) \right],\ k = 1, \ldots, n.$

Замечание.

Для произвольной нижней вероятности g представление (1) для коэффициентов оптимальной вероятности будет неверным.

Укажем на одну вероятностную интерпретацию коэффициентов α_k, $k = 1, \ldots, n$. Для этого, учитывая свойство 3) симметричной части ρ_g, запишем коэффициент α_k в виде

$$\alpha_k = \sum_{A \subseteq X} \frac{\rho_g(A)}{2^{|X|-1}} \cdot \frac{1 + 2|X| \eta_k(A) - 2|A|}{|X|}.$$

Пусть $p(A) = 2^{1-|X|} \cdot \rho_g(A)$, $q_k(A) = (1 + 2|X| \eta_k(A) - 2|A|) / |X|$. Тогда имеем случайную величину Q_k, принимающую значения $q_k(A)$ ($A \subseteq X$) с вероятностями $p(A)$ ($A \subseteq X$) и $\alpha_k = \mathbf{M}[Q_k]$.

Список литературы

1. Хьюбер Дж. П. Робастность в статистике. М. Мир, 1984.

2. Murofushi T., Sugeno M. An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure // Fuzzy Sets and Systems. 1989. № 29.

3. Данилов В. И. Лекции по теории игр. М. Российская экономическая школа, 2002.

4. Chateauneuf A., Jaffray J.Y. Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion // Mathematical Social Sciences. 1989. № 17.

5. Smets P. Belief functions and transferable belief model // http://ippserv.rug.ac.be.

6. Denneberg D., Grabisch M. Interaction transform of set function over a finite set // Information Sciences, 121, 1999, pp.149-170.

7. Denneberg D. Non-additive measure and integral, basic concepts and their role for applications // In M. Grabisch, T. Murofushi, M. Sugeno (eds.): Fuzzy Measures and Integrals - Theory and Applications. Studies in Fuzzyness and Soft Computing 40. Heidelberg. Physical-Verlag, 2000.

О выразимости константных автоматов суперпозициями

Летуновский А. А.

МГУ им М.В. Ломоносова, механико-математический факультет.

e-mail: letunovs@yandex.ru

Рассматриваются задачи выразимости и А – выразимости константных автоматных функций относительно суперпозиции. Показано, что нет алгоритма определения по конечному базису мощности множества выразимых (А-выразимых) через него констант. Приводятся достаточные условия, при которых мощности множества выразимых (А-выразимых) констант конечны, и достаточные условия, при которых мощности множества выразимых (А-выразимых) констант бесконечны.

Введение. Известно, что решение задачи о полноте относительно операции суперпозиции и обратной связи для систем автоматных функций наталкивается на существенные трудности. Так в работе [1] установлена континуальность всякой критериальной системы для этой задачи, в работе [2] алгоритмическая неразрешимость задачи о полноте для конечных систем автоматных функций, а в работе [3] - алгоритмическая неразрешимость задачи об А-полноте. Вместе с тем, для систем автоматов, содержащих все истинностные функции, задача о полноте [4,5] и А-полноте [6] алгоритмически разрешимы.

Для автоматов с операцией суперпозиции наибольший интерес представляет задача выразимости, так как все полные системы в это алгебре бесконечны. В данной работе изучена выразимость константных автоматных функций. Кратко И.М. в работе[1] доказал, что задача выразимости константных автоматных функций и задача пустоты множества выразимых автоматных функций алгоритмически неразрешимы.

Основные результаты.

Пусть $E_k = \{0,1,...,k-1\}$, функции вида $g : E_k^n \to E_k$ называются функциями k-значной логики, их множество обозначается через P_k. Пусть E_k^∞ - множество всех сверхслов вида $a(1)a(2)...$, где $a(j) \in E_k$, $j = 1,2,...$, E_k^τ - множество всех слов $a(1)...a(\tau)$ длины τ. Через N обозначим множество натуральных чисел. Пусть $f : (E_k^\infty)^n \to E_k^\infty$ – автоматная функция (*a*-функция), т.е. она задается рекуррентно соотношениями (1)

$$\begin{cases} q(1) = q_1, \\ q(t+1) = \varphi(q(t), a_1, ..., a_n), \\ b(t) = \psi(q(t), a_1, ..., a_n) \end{cases} \tag{1}$$

где $q \in Q = \{q_1, q_2, ..., q_r\}$. Параметр q называется состоянием *a*-функции f, q_1 ее начальным состоянием, буквы $a = (a_1, a_2, ..., a_n)$ и b называются входной и выходной буквами, а сверхслова $a(1)a(2)...$ и $b(1)b(2)...$ - входным и выходным сверхсловами, соответственно. Функции φ и ψ называются функциями переходов и выходной функцией, соответственно, а шестерка $(E_k^n, Q, E_k, \varphi, \psi, q_1)$ – автоматом, порождающим функцию f. Класс всех *a*-функций обозначим через P. В этом классе обычным образом введем операции суперпозиции. Для суперпозиции будем использовать модификации операций из [7]:

$$\begin{cases} (\eta f)(x_1, x_2, ...x_n) = f(x_2, x_3, ...x_n, x_1) \\ (\varepsilon f)(x_1, x_2, ...x_n) = f(x_2, x_1, x_3, ...x_n) \\ (\omega f)(x_1, x_2, ...x_{n-1}) = f(x_1, x_1, x_2, ...x_{n-1}) \\ (\delta f)(x_1, x_2, ...x_{n+1}) = f(x_2, x_3, ...x_{n+1}) \\ (f * g)(x_1, x_2, ...x_{m+n+1}) = f(g(x_1, ...x_m), x_{m+1}, ...x_{m+n-1}) \end{cases}$$

Пусть $M \subseteq P$, обозначим через $[M]$ - множество а-функций, получающихся из M с помощью операций суперпозиции. Пусть $\tau \in N$, f - некоторая автоматная функция, обозначим через $f^\tau : (E_k^\tau)^n \to (E_k^\tau)$ ограничение этой функции на множество слов длины τ. Скажем, что а-функции $f(x_1, ..., x_n)$ и $g(x_1, ..., x_n)$ - τ-равны, если $f^\tau = g^\tau$. Обозначим через $[M]_\tau$ - множество всех а-функций, τ-равных получающимся из M с помощью суперпозиции , пусть $[M]_A = \bigcap_{\tau=1}^{\infty} [M]_\tau$, назовем $[M]_A$ А-замыканием множества M.

Автоматная функция f - называется константной, если для любого входного сверхслова $a(1)a(2)...$ ее выходное сверхслово $f(a(1)a(2)...) \equiv b(1)b(2)... = \beta$.

Когда это не приводит к недоразумению, мы будем отождествлять константную автоматную функцию f с ее выходным сверхсловом и обозначать той же буквой. Класс всех константных автоматных функций обозначим через K.

Пусть $K' \subseteq K$, обозначим через $A(K')$ - множество сверхслов , которые получаются на выходе автомата A при подаче сверхслов из K', для конечного множества автоматов $M = \{A_1, ..., A_n\}$ обозначим $M(K') = \bigcup_{i=1}^{n} A_i(K')$. Для натурального τ через $K']_\tau$ обозначим множество начал длины τ слов из K'. Для автомата A определим последовательность множеств:

$L_0 = E_k$, $L_1(A) = P_k(L_0 \cup A(L_0))$, . . . $L_{i+1}(A) = P_k(L_i \cup A(L_i))$, ,

обозначим $L(\mathrm{A}) = \bigcup_{i=0}^{\infty} L_i(\mathrm{A})$, $L(M) = \bigcup_{i=1}^{n} L(A_i)$.

Для $i \neq j$ через $A_{ij} \subset K$ обозначим множество сверхслов $a(1)a(2)\ldots$, у которых $a(i) = a(j)$. Скажем, что автомат сохраняет множество A_{ij} , если $\mathrm{A}(A_{ij}) \subseteq A_{ij}$, в противном случае будем говорить, что автомат отличает моменты времени i и j . Будем говорить, что M сохраняет A_{ij} , если $\forall k = 1..n$ A_k сохраняет A_{ij} .

Определим для автомата A последовательность множеств состояний:

$Q_0 = \{q_1\}$, $Q_1 = \{\varphi(q_1, a) \mid a \in E_k^n\}$, … $Q_{i+1} = \{\varphi(q_i, a) \mid q_i \in Q_i, a \in E_k^n\}$.

Это периодическая последовательность, пусть d - ее предпериод, а ρ_0 -период, r - число состояний автомата, понятно, что $\rho_0 < 2^r$, $d < 2^r$. Обозначим через $\rho(\mathrm{A}) = d + \rho_0$. $\rho(M) = \sum_{i=1}^{n} \rho_0(A_i) + \prod_{i=1}^{n} d(A_i)$

Мы будем рассматривать следующие задачи: по конечному множеству M и $\beta \in K$ проверить, верно ли, что $\left|[M] \cap \mathrm{K}\right| = \infty$, $\left|[M]_{\mathrm{A}} \cap \mathrm{K}\right| = \infty$, которые назовем задачей бесконечности множества выразимых (А-выразимых) констант. Имеют место следующие результаты.

Теорема 1. *Задача бесконечности множества выразимых констант, алгоритмически неразрешима.*

Теорема 2. *Задача бесконечности множества А-выразимых констант алгоритмически неразрешима.*

Теорема 3. *Если для некоторых* i *и* j $\mathrm{i, j} < \rho(\mathrm{M})$ *множества* $A_{i,i+j}, A_{i+1,i+j+1}, \ldots, A_{i+s,i+j+s}$, *где* $s = j \cdot |Q|$, *сохраняются* M , *тогда* $L(M)$ - *конечно.*

Теорема 4. *Если* $[M] \supseteq P_2$ *и для всех* $i, j < \rho(M)$, $i \neq j$ *множество* M *отличает моменты времени* i *и* j , *тогда* $L(M)$ - *бесконечно.*

Теорема 5. *Если* $\left|[M] \cap \mathrm{K}\right| \neq \varnothing$, *то* $\left|[M] \cap \mathrm{K}\right| < \infty$ *тогда и только тогда когда* $\left|[M]_{\mathrm{A}} \cap \mathrm{K}\right| < \infty$

Автор выражает благодарность академику Кудрявцеву В. Б. и проф. Бабину Д. Н. за ценные замечания и внимание к работе.

Список литературы

1. Кудрявцев В. Б., О мощностях множеств предполных классов некоторых функциональных систем, связанных с автоматами, ДАН СССР т.151,N3,1963, с.493-496.

2. Кратко М. И., Алгоритмическая неразрешимость проблемы распознавания

полноты для конечных автоматов, ДАН СССР, 1964, т.155, N 1, с.35--37.

3. Буевич В. А., Об алгоритмической неразрешимости распознавания А-полноты для о.д.-функций, Математические заметки, том 12, номер 6, 1972, с.687-697.

4. Бабин Д. Н., Разрешимый случай задачи о полноте автоматных функций,

Дискретная математика, том 4, 1992, выпуск 4, с.41-56, Наука, Москва.

5. Бабин Д. Н., О классификации автоматных базисов Поста по разрешимости свойств полноты и А-полноты, ДОКЛАДЫ АКАДЕМИИ НАУК, N 4, Т.367, 1999 с. 439-441

6. Буевич В. А., Условия А-полноты для автоматов, изд. МГУ, 1986 г.

7. Мальцев А. И., Итеративные алгебры и многообразие Поста, Алгебра и логика, 1966, т.5, N2, с.5--24.

8. Кудрявцев В. Б., Алешин С. В., Подколзин А. С., Введение в теорию автоматов, Наука, М., 1985.

Решение автоматных уравнений

Лялин И. В.,
МГУ им. М. В. Ломоносова

Постановка задачи

Пусть дана произвольная автоматная схема S, построенная из конечных автоматов с помощью операций суперпозиции и обратной связи. Пусть в S выделено произвольное множество автоматов $x_1, x_2, \ldots, x_n$. Автоматы из этого множества будем называть *свободными позициями*, а остальные автоматы в схеме S – *фиксированными*. Схему S со свободными позициями будем обозначать так: $S(x_1, \ldots, x_n)$. Пусть вместо автоматов x_i разрешается подставлять в схему S любые другие автоматы с тем же числов входов и выходов. То есть если у автомата x_1 a_1 входов и b_1 выходов то вместо него разрешается подставлять любой автомат c_1 у которого a_1 входов и b_1 выходов, при этом i-тый вход автомата u подключается в схему туда где раньше был подключен i-тый вход автомата x_1, а j–ый выход автомата u подключается в схему туда где раньше был подключен j-тый выход автомата x_1. Пусть каждый x_i заменен таким образом на какой-то автомат c_i. Тогда схема S станет эквивалентна некоторому автомату h. Будем это записывать так: $S(c_1, \ldots, c_n) = h$. Основная задача, рассатриваемая в данной работе, ставится следующим образом: дана схема со свободными позициями $S(x_1, \ldots, x_n)$ и автомат h, требуется найти такие автоматы $c_1, \ldots, c_n$, чтобы $S(c_1, \ldots, c_n) = h$. Таким образом, функционирование автоматов x_i нам не важно, они используются только как позиции, вместо которых подставляются автоматы c_i. Можно даже сказать что x_i – это переменные со значением в множестве всех автоматов, имеющих определенное число входов и выходов. А $S(x_1, \ldots, x_n) = h$, таким образом, есть уравнение с n неизвестными, которое требуется решить. Будем такие уравнения называть *автоматными*.

Основная задача: Научиться решать автоматные уравнение с n неизвестными $S(x_1, \ldots, x_n) = h$.

Основные полученные результаты:

Теорема 1. Автоматные уравнения с одной неизвестной алгоритмически разрешимы.

Теорема 2. Автоматные уравнения с тремя и более неизвестными алгоритмически неразрешимы.

Определение 1. h_0 – константный детерминированный автомат, всегда на выходе выдающий 0.

Утверждение 1 (упрощение уравнения). Для любой схемы-шаблона S и любой о.-д. функции h существует такая схема-шаблон S' что множество решений уравнения $S(x_1, x_2, \ldots, x_n) = h$ совпадает со множеством решений уравнения $S'(x_1, x_2, \ldots, x_n) = h_0$

Определение 2. *Недетерминированным автоматом (НДА)* называется следующая пятерка: $\langle A, B, U, u^0, \rho\rangle$, где

A – входной алфавит (конечное множество).

B – выходной алфавит (конечное множество).

U – *множество состояний* НДА (конечное множество).

$u^0 \subseteq U$ – *множество начальных состояний* (состояния из этого множества называются *начальными*). u^0 может быть пустым.

$\rho : U \times A \to 2^{U \times B}$, где $2^{U \times B}$ – множество всех подмножеств множества $U \times B$. ρ называется *функцией перехода* .

Определение 3. НДА, у которого множество состояний пусто, будем называть *пустым*.

Пустой НДА – это вырожденный, но важный случай. Как пустое множество.

Определение 4. Будем говорить что ДА $z = \langle A, B, Q, q^0, \phi, \psi\rangle$ *вкладывается* в НДА $N = \langle A, B, U, u^0, \rho\rangle$ если существует такое отображение $\omega : Q \to 2^U$, что:

1) $\omega(q^0) \cap u^0 \neq \emptyset$

2) Пусть между двумя состояниями $q_1, q_2 \in Q$ есть переход a/b, то есть $q_2 = \phi(q_1, a)$ и $b = \psi(q_1, a)$. Тогда

$$\forall u_1 \in \omega(q_1)\ \exists u_2 \in \omega(q_2) \text{ т.ч. } (u_2, b) \in \rho(u_1, a)$$

Про отображение ω будем говорить, что оно *вкладывает* z в N.

Утверждение 2. Пусть автомат $z_1 = \langle A, B, Q_1, q_1^0, \phi_1, \psi_1\rangle$ отображением w вложим в НДА $N = \langle A, B, U, u^0, \rho\rangle$ и автомат $z_2 = \langle A, B, Q_2, q_2^0, \phi_2, \psi_2\rangle$ эквивалентен автомату z_1. Тогда z_2 тоже вложим в N.

Утверждение 3. Множество автоматов, вложимых в НДА N пусто тогда и только тогда когда N пуст.

Одна неизвестная с обратными связями

Обозначения:

A – входной алфавит схемы S.
$B = \{0, 1\}$ – выходной алфавит схемы S.
l – количество фиксированных автоматов в схеме S.
$K_i = \langle A_i, B_i, Q_i, q_i^0, \phi_i, \psi_i \rangle$ – i-ый фиксированный автомат в схеме S $(1 \le i \le l)$.
A' – входной алфавит свободной позиции.
B' – выходной алфавит свободной позиции.
F – множество функций $f : A' \to B'$.

1. Построим по схеме – шаблону S граф $\overline{G}_1(S)$. $\overline{G}_1(S)$ – ориентированный и у него каждая дуга будет иметь метку из множества $A \times B \times A'$. Вершинами этого графа будет множество $Q_1 \times Q_2 \times \ldots \times Q_l \times F$. Вершины вида $(q_1^0, \ldots, q_l^0, f)$ являются выделенными, где $f \in F$. Будем говорить, что S находится в состоянии $q = (q_1, \ldots, q_l)$, если автомат K_i находится в состоянии q_i. Пусть S находится в состоянии $q = (q_1, \ldots, q_l)$. Пусть $f \in F$ – функция выхода свободной позции в этот момент времени. Пусть на вход S подали $a \in A$. Пусть на вход свободной позиции пришло при этом $a' \in A'$, на выходе свободной позиции получили $b' = f(a') \in B'$, на выходе S получили $b \in B$ и i–ый фиксированный автомат перешёл в состояние r_i. Тогда и только тогда для любого $f_2 \in F$ из вершины $q = (q_1, \ldots, q_l, f)$ в вершину $r = (r_1, \ldots, r_l, f_2)$ идет дуга с пометкой (a, b, a').

2. Построим $\overline{G}_2(S)$. Он получается из $\overline{G}_1(S)$ путем выкидывания некоторых вершин. Рассмотрим произвольную вершину $q = (q_1, \ldots, q_l, f)$ графа $\overline{G}_1(S)$. Зная вершину мы знаем в каких состояниях находятся все фиксированные автоматы, а следовательно и функции которые они реализуют в этих состояниях. Свободная позиция в этом состоянии реализует функцию f. Этого достаточно для проверки корректности в данный момент времени всех обратных связей. Если хотя бы одна из них не корректна (т.е. функция выхода зависит от входа, с которым выход соединен обратной связью), то такое состояние выбрасываем из $\overline{G}_1(S)$. Оставшийся граф назовем $\overline{G}_2(S)$.

3. Построим $\overline{G}_3(S)$. Для этого проведём с $\overline{G}_2(S)$ следующую процедуру. Назовём дугу графа $\overline{G}_2(S)$ неправильной, если она помечена меткой вида $(a, 1, a')$. Выбросим из $\overline{G}_2(S)$ все неправильные дуги. Получим граф $\overline{G}'_2(S)$. Вершину графа $\overline{G}'_2(S)$ назовём неправильной, если $\exists a \in A$ такое что множество дуг выходящих из данной вершины и помеченных метками вида $(a, 0, a')$ пусто. Выкинем из $\overline{G}'_2(S)$ все неправильные вершины. При выкидывании вершины выкидываются также все дуги выходящие из неё или входящие в неё. После этого какие–нибудь другие вершины могут стать неправильными. Выкидываем и их тоже. Продолжаем выкидывание до тех пор пока в графе не останется неправильных вершин. Полученный граф назовём $\overline{G}''_2(S)$. Тогда $\overline{G}_3(S)$ это подграф графа $\overline{G}''_2(S)$, содержащий все его начальные вершины и все вершины до которых можно дойти из начальных по дугам графа $\overline{G}''_2(S)$.

Заметим что все дуги в $\overline{G}_3(S)$ – правильные, т.е. вида $(a, 0, a')$. Следовательно, можно убрать второй элемент, оставив только первый и третий в качестве метки дуги. Будем считать что метки дуг в $\overline{G}_3(S)$ принадлежат множеству $(A \times A')$. Множество вершин графа $\overline{G}_3(S)$ обозначим Q.

Определение 5. Пусть в схеме-шаблоне $S(x)$ на свободной позиции стоит автомат z. Тогда $q = (q_1, \ldots, q_l; f) \in Q$ есть *сопутствующее* состояние автомата $S(z)$ в какой–то момент времени если фиксированный автомат K_i в этот момент времени находится в состоянии q_i а f – выходная функция автомата z в этот момент времени.

Утверждение 4. Пусть $S(z) = h_0$. Тогда $S(z)$ в любой момент времени находится в сопутствующем состоянии, входящим в Q.

4. Построим граф $\overline{G}_4(S)$.

Множество вершин графа $\overline{G}_4(S)$ – множество всех подмножеств множества Q, у всех элементов которых на свободной позиции стоит одна и та же функция. Обозначим его U. Функцию, стоящую на свободной позиции в состоянии $u \in U$, обозначим $\psi(u)$. Также в U входят пустые подмножества со всевозможными функциями из F. Будем обозначать их $\emptyset_f$, где $f \in F$ – функция, поставленная в соответствие данному пустому подмножеству.

Обозначим $\chi^1(q, a')$ множество всех вершин графа $\overline{G}_3(S)$ в которые из q идут дуги с меткой вида $(\ldots, a')$.

$\chi^2(u, a') \stackrel{\text{def}}{=} \bigcup\limits_{q \in u} \chi^1(q, a')$. В частности, $\chi^2(\emptyset_f, a') = \emptyset$.

$\overline{G}_4(S)$ – ориентированный и у него каждая дуга имеет метку из множества A'. Из вершины u в вершину u' тогда и только тогда идет дуга с меткой a' когда существует такая $f \in F$ что

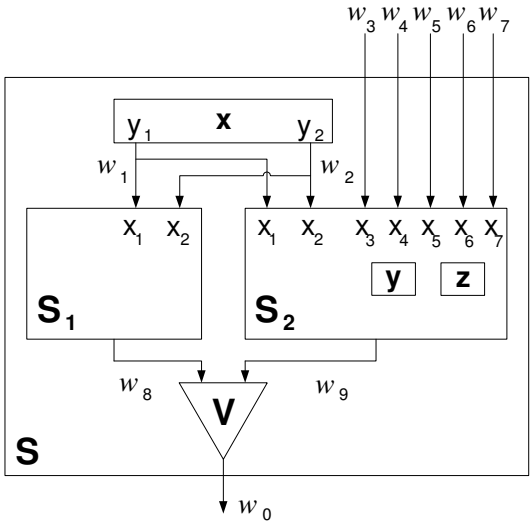


Рис. 1:

$u' = (\chi^2(u, a'), f)$. В частности, если $\chi^2(u, a') = \emptyset$ то из u дуги с пометкой a' идут во все вершины вида $\emptyset_f$ графа $\overline{G}_4(S)$. Выделенными вершинами будут все вершины вида $\{(q_1^0, \ldots, q_l^0; f)\}$. $\overline{G}_4(S)$ является ПНДА, если функцией выхода в состоянии u считать $\psi(u)$.

Утверждение 5 (решение уравнения с одной неизвестной). Автомат z являетя решением автоматного уравнения $S(x) = h_0$ тогда и только тогда когда z вложим в $\overline{G}_4(S)$.

Уравнение с тремя неизвестными

Задача распознавания сочетаемости. Пусть $E_m = \{0, 1, \ldots, m-1\}$ – конечный алфавит. Пусть $A_0, \ldots, A_{p-1}$ и $B_0, \ldots, B_{p-1}$ – слова в этом алфавите. Данный набор слов назовём системой Поста. Система Поста называется сочетаемой, если существуют такие $i_1, \ldots, i_r \in E_p$ что слово $A_{i_1}A_{i_2}...A_{i_{r-1}}A_{i_r}$ совпадает со словом $B_{i_1}B_{i_2}...B_{i_{r-1}}B_{i_r}$. Данные $i_1, \ldots, i_r$ есть решение системы Поста. Количество индексов в решении не должно быть нулевым. Пост доказал что для любого $m > 1$ задача распознавания сочетаемости алгоритмически неразрешима.

Утверждение 6. Для любой системы Поста можно конструктивно построить автоматное уравнение с тремя неизвестными которое будет иметь решение если и только если имеет решение соответствующая система Поста. Это доказывает теорему 2.

Итак, определим как по системе Поста строится автоматное уравнение и докажем, что система Поста имеет решение если и только если имеет решение построенное автоматное уравнение.

Пусть дана произвольная система Поста Π (то есть алфавит E_m и множество слов $A_0, \ldots, A_{p-1}$ и $B_0, \ldots, B_{p-1}$ в нем). Построим по этой системе схему $S(x, y, z)$, изображенную на рис. 1. Эта схема строится таким образом, чтобы соответствующее ей автоматное уравнение $S(x, y, z) = h_0$ имело решение если и только если система Поста разрешима. h_0 – константный автомат, всегда на выходе выдающий 0. Схема S состоит из свободной позиции x, двух подсхем S_1, S_2 и дизъюнкции. У x два выхода (будем называть их y_1 и y_2) и ни одного входа. Сверхслово на выходе y_1 будем обозначать w_1, а на выходе y_2 – w_2. У S_1 два входа x_1, x_2 и один выход. Вход x_1 соединен с выходом y_1 свободной позиции x, а вход x_2 – с выходом y_2 свободной позиции x. Сверхслово на выходе S_1 будем обозначать w_8. У S_2 семь входов x_1, x_2, x_3, x_4, x_5, x_6, x_7 и один выход. Вход x_1 соединен с выходом y_1 свободной позиции x, вход x_2 – с выходом y_2 свободной позиции x, а остальные входы являются входами всей схемы S. Пусть сверхслово на входе x_3 – w_3, x_4 – w_4, x_5 – w_5, x_6 – w_6, x_7 – w_7. Обозначим w_9 сверхслово на выходе S_2. Внутри S_2 находятся свободные позиции y и z. Выход подсхемы S_1 и выход подсхемы S_2 соединены со входами дизъюнкции. Выход дизъюнкции является выходом схемы S. Обозначим w_0 сверхслово на выходе схемы S. Чтобы $w_0 = (00\ldots)$ необходимо чтобы $w_8 = (00\ldots)$ и $w_9 = (00\ldots)$.

Определение 6. Множество сверхслов в алфавите E_{p+2} следующего вида:

$$(i_1 \underbrace{p\ p...p}_{\geq 0}\ i_2 \underbrace{p\ p...p}_{\geq 0}\ \ldots\ i_r \underbrace{p\ p...p}_{\geq 0}\ \overline{p+1}\ \overline{p+1}\ \ldots)$$

где $0 \leq i_k \leq p-1$, будем обозначать U_p. Пусть $P \in U_p$. Количество букв в P отличных от p и $p+1$ будем обозначать $len(P)$. Их может быть счетное число, в этом случае $len(P) = \infty$. k-ую букву,

отличную от p и $p+1$, будем обозначать $lt(P,k)$, $1 \le k \le len(P)$. Позиция, на которой стоит k-ая буква, отличная от p и $p+1$, обозначим $pl(P,k)$. Положим $pd(P,k)$ $(1 \le k \le len(P))$ – количество букв p, стоящих после k-ой буквы, отличной от p и $p+1$. $pd(P,k) \ge 0$. Обозначим $es(P)$ длину самого длинного префикса сверхслова P, в котором нет буквы $p+1$. Слово, составленное буквами $lt(P,k)$, $1 \le k \le len(P)$, обозначим $wd(P)$. То есть $wd(P) = lt(P,1)lt(P,2)\dots lt(P,len(P))$. Таким образом, слово P выглядит следующим образом:

$$\underbrace{(lt(P,1)\underbrace{p\ p\ \dots\ p}_{pd(P,1)}\ lt(P,2)\underbrace{p\ p\ \dots\ p}_{pd(P,2)}\ \dots\ lt(P,len(P))\ \underbrace{p\ p\ \dots\ p}_{pd(P,len(P))}}_{es(P)}\ \overline{p+1}\ \overline{p+1}\ \dots)$$

Определение 7. Множество всех таких сверхслов $P \in U_p$, что $pd(P,k) = |A_{lt(P,k)}| - 1$, $1 \le k \le len(P)$, будем обозначать U_A. Множество всех таких $P \in U_p$, что $pd(P,k) = |B_{lt(P,k)}| - 1$, $1 \le k \le len(P)$, будем обозначать U_B.

Определение 8. Множество всех таких сверхслов $P \in U_p$, что $len(P) = \infty$, будем обозначать U_p^∞. Множество $U_p \backslash U_p^\infty$ обозначим U_p^l.

Блоки S_1 и S_2 строятся таким образом, что верны следующие утверждения:

Утверждение 7. $w_8 = (00\dots)$ тогда и только тогда когда $w_1 \in U_A$, $w_2 \in U_B$, $len(w_1) > 0$, $len(w_2) > 0$ и
$A_{lt(w_1,1)}A_{lt(w_1,2)}\dots A_{lt(w_1,len(w_1))} = B_{lt(w_2,1)}B_{lt(w_2,2)}\dots B_{lt(w_2,len(w_2))}$

Утверждение 8. Если $w_1 \in U_p^\infty$ и $w_2 \in U_p^\infty$, то найдутся такие w_3, w_4, w_5, w_6, w_7 что $w_9 \ne (00\dots)$.

Утверждение 9. Пусть $w_1, w_2 \in U_p^l$ и $es(w_1) = es(w_2)$. В этом случае $wd(w_1) = wd(w_2)$ тогда и только тогда когда существуют такие Y и Z, подставив которые на свободные позиции y и z, получим $w_9 = (00\dots)$ для любых w_3, w_4, w_5, w_6, w_7.

Используя эти утверждения, можно доказать следующее утверждение, которое доказывает утверждение 6.

Утверждение 10. Система Поста Π имеет решение если и только если имеет решение уравнение $S(x,y,z) = h_0$.

Доказательство. Пусть Π имеет решение $i_1, \dots, i_r$. Возьмем в качестве X такой константный автомат, чтобы $w_1 \in U_A$, $w_2 \in U_B$ и $wd(w_1) = wd(w_2) = i_1 i_2 \dots i_r$. Очевидно, что такой автомат существует. Причем $es(w_1) = |A_{i_1} \dots A_{i_r}| = |B_{i_1} \dots B_{i_r}| = es(w_2)$. Тогда по утверждению 7 $w_8 = (00\dots)$. А по утверждению 9 существуют такие автоматы Y и Z что $w_9 = (00\dots)$. Значит, $S(X,Y,Z) = h_0$.

Пусть существуют такие автоматы X, Y и Z что $S(X,Y,Z) = h_0$. Значит $w_8 = (00\dots)$ и $w_9 = (00\dots)$. Согласно утверждению 7 отсюда следует что $w_1 \in U_p$, $w_2 \in U_p$ и

$$A_{lt(w_1,1)}A_{lt(w_1,2)}\dots = B_{lt(w_2,1)}B_{lt(w_2,2)}\dots \qquad (1)$$

По утверждению 8 следует что $w_1 \in U_p^l$ или $w_2 \in U_p^l$. Понятно, что если одно сверхслово из пары (w_1, w_2) лежит в U_p^l, то и второе находится там же, потому что иначе, ввиду конечности всех слов B_i и A_i, не может выполняться (1). А если (1) выполняется, то $es(w_1) = es(w_2)$. Отсюда по утверждению 9 следует что $wd(w_1) = wd(w_2)$. Значит, $wd(w_1)$ является решением система Поста Π. Его длина больше нуля, т.к. по утверждению 7 $len(w_1) > 0$. ■

Список литературы

1. Яблонский С. В., Введение в дискретную математику. Издательство "Наука", Москва, 1979

2. Трахтенброт Б. А., Барздинь Я. М., Конечные автоматы (поведение и синтез). Издательство "Наука", Москва, 1970

3. Кудрявцев В. Б., Алешин С. В., Подколзин А. С., Введение в теорию автоматов. Издательство "Наука", Москва, 1985

4. Подколзин А. С., Ушчумлич Ш. М., О решении систем автоматных уравнений. Дискретная Математика, том 2 выпуск 1, 1990

5. Лялин И. В., О решении автоматных уравнений. Дискретная Математика, том 16 выпуск 2, 2004

Коммуникационная сложность протоколов доступа к данным без раскрытия запроса

Майлыбаева Г. А.,
кафедра МаТИС механико-математического факультета МГУ им. М. В. Ломоносова
E-mail: guliam@yandex.ru

Рассмотрим протокол с $k+1$ участником: пользователем и k несообщающимися серверами ($k \geq 1$), причем каждый из серверов хранит один и тот же булев вектор $x = (x_0, \ldots, x_{n-1})$ длины n — базу данных. Пользователь желает узнать значение i-го бита x_i этого вектора так, чтобы номер бита i не стал известен ни одному из серверов. Протокол, который позволяет это делать, называется протоколом доступа к данным без раскрытия запроса или PIR-протоколом и определяется следующим образом.

Для любого натурального n обозначим $E_n = \{0, \ldots, n-1\}$. Пусть k, n, s, m, $p^0, \ldots, p^{k-1}$ — натуральные числа, $p = p^0 + \ldots + p^{k-1}$. Пусть на множестве $B = \{(i, r),\ i \in E_n,\ r \in E_s\}$ задано вероятностное пространство $\langle B, 2^B, \mathrm{P}\rangle$, где $\mathrm{P}(i, r) = \frac{1}{n \cdot s}$, для любых $i \in E_n$, $r \in E_s$. Тогда (k, n, s, m, p) *PIR-протоколом* называется набор из $k+2$ функций $I = \langle Q, A^0, \ldots, A^{k-1}, R\rangle$, где $Q, A^0, \ldots, A^{k-1}, R$ некоторые отображения $Q : E_k \times E_n \times E_s \to E_m$, $A^j : E_m \times \{0,1\}^n \to \{0,1\}^{p^j}$, $j \in E_k$, $R : E_n \times E_s \times \{0,1\}^p \to \{0,1\}$, такие, что выполнено 2 условия:

корректности: для любых $i \in E_n$, $r \in E_s$ выполнено

$$R(i, r, A^0(Q(0, i, r), x), \ldots, A^{k-1}(Q(k-1, i, r), x)) = x_i.$$

защищенности: для любых $q \in E_m$, $t \in E_k$, $i, j \in E_n$ выполнено

$$\mathrm{P}(Q(t, i, r) = q) = \mathrm{P}(Q(t, j, r) = q).$$

Содержательно протокол $I = \langle Q, A^0, \ldots, A^{k-1}, R\rangle$ состоит из следующих шагов:

- Пользователь U, имея запрос i, вырабатывает случайное число $r \in E_s$, для каждого $j \in E_k$ вычисляет $q^j = Q(j, i, r)$ и посылает q^j j-му серверу S_j.
- Каждый сервер S_j, $j \in E_k$, вычисляет $a^j = (a^j_0, \ldots, a^j_{p^j-1}) = A^j(x, q^j)$ и посылает вектор a^j пользователю.
- U вычисляет $x_i = R(i, r, a^0, \ldots, a^{k-1})$.

Величина $C(I) = k]\log_2 m[+p$ называется *коммуникационной сложностью протокола* I. $C(I)$ — число бит, переданных в процессе протокола.

Условие корректности гарантирует, что пользователь получит нужный бит базы данных, а условие защищенности — что ни один из серверов по вектору q, который он получил, не сможет понять какой бит интересует пользователя. Предполагается, что всем участникам протокола и пользователю и серверам известны функции запросов, ответов и реконструирующая. Но серверам не известно случайное число r и разумеется не известен номер бита i.

Основной целью исследований в этой области является построение для заданного количества серверов k, длины базы данных n и максимального значения датчика случайных чисел s PIR-протокола с минимальной коммуникационной сложностью.

PIR-протокол, у которого коммуникационная сложность больше либо равна длине базы данных, будем считать вырожденным.

Нами были найдены границы вырожденности PIR-протокола по основным его параметрам.

Так показано, что если количество серверов $k = 1$, то нельзя построить невырожденный PIR-протокол, но уже при $k = 2$ существует невырожденный PIR-протокол.

Показано, что если m — мощность множества значений функции запросов PIR-протокола, то при $m \leq 1$ не существует и при $m \geq 2$ существует невырожденный PIR-протокол.

Показано, что если s — мощность множества значений датчика случайных чисел, используемого в PIR-протоколе, то при $s \leq 1$ не существует и при $s \geq 2$ существует невырожденный PIR-протокол.

Степенью существенности булевой функции $f(x_1, \ldots, x_l)$ назовем число переменных, от которых она существенно зависит, и обозначим его через $S(f)$. Степенью существенности булевой вектор-функции $F(x_1, \ldots, x_l) = (f_1(x_1, \ldots, x_l), \ldots, f_t(x_1, \ldots, x_l))$ назовем число $S(F) = \max\limits_{1 \leq j \leq t} S(f_j)$.

Пусть $A^j(q)(x) = A^j(q, x) = (A^j_0(q, x), \ldots, A^j_{p^j-1}(q, x)), \forall j \in E_k$; $R_{i,r}(a^0, a^1) = R(i, r, a^0, a^1), \forall i \in E_n, r \in E_s$.

Степенью существенности функции ответов j-го сервера $A^j : E_s \times \{0,1\}^n \to \{0,1\}^{p^j}$, $j \in E_2$, назовем число $S(A^j) = \max\limits_{q \in E_s} S(A^j(q))$.

Степенью существенности реконструирующей функции назовем число $S(R) = \max\limits_{i \in E_n, r \in E_s} S(R_{i,r})$.

Показано, что если в ответах серверов содержатся только открытые биты базы данных, т.е. если каждый бит ответов существенно зависит только от одного бита базы данных, то не существует невырожденного PIR-протокола, и наоборот, если хотя бы один сервер имеет функцию запросов такую, что биты его ответа существенно зависят хотя бы от двух битов базы данных, то невырожденный PIR-протокол существует.

По поводу реконструирующей функции показано, что если она существенно зависит только от одного бита из ответов серверов, то не существует невырожденного PIR-протокола, но если реконструирующая функция существенно зависит хотя бы от двух битов из ответов серверов, невырожденный PIR-протокол существует.

Обозначим через $\mathcal{I}(2,n,s)$ класс всех $(2,n,s,p)$ PIR-протоколов, где $p > 0$. Пусть $\mathcal{A}$ — некоторое множество PIR-протоколов. Тогда обозначим

$$C(2,n,s,\mathcal{A}) = \min\{C(I) : I \in \mathcal{A} \cap \mathcal{I}(2,n,s)\}.$$

Для любого натурального d обозначим через $\mathcal{A}_{d,2}$ множество всех PIR-протоколов таких, что степень существенности функции ответов каждого сервера не превосходит d, а степень существенности реконструирующей функции не превосходит 2.

Теорема 1. *Для любых натуральных n, s верно*

$$C(2,n,s,\mathcal{A}_{2,2}) \le 2]\log_2 s[+\frac{s+1}{2s}n + n \mod s^2.$$

Пусть функция $\mathrm{sign} : \mathbb{N} \cup \{0\} \to \{0,1\}$ такая что

$$\mathrm{sign}(n) = \begin{cases} 0, & \text{если } n = 0, \\ 1, & \text{если } n \ge 1. \end{cases}$$

Теорема 2. *Для любых натуральных n, s верно*

$$C(2,n,s,\mathcal{A}_{2,2}) \ge min\{n, 2]\log_2 s[+\frac{s+1}{2s}n - \frac{s+3}{3}(sign(n') - \frac{n'}{2s})\},$$

где $n' = n \mod 2s$.

Следствие 1. *Для любых натуральных n, s таких что n кратно $2s^2$, верно*

$$C(2,n,s,\mathcal{A}_{2,2}) = 2]\log_2 s[+\frac{s+1}{2s}n.$$

Теорема 3. *Для любого натурального n и любого натурального s такого, что $\log_2 s$ — целое, верно*

$$C(2,n,s,\mathcal{A}_{\log_2 s,2}) \le 2\log_2 s + 2]\frac{n}{\log_2 s}[.$$

Теорема 4. *Для любых натуральных n, s, d таких, что $s \ge n$, верно*

$$C(2,n,s,\mathcal{A}_{d,2}) \ge 2]\log_2 s[+\frac{n+1}{d+1} + \frac{n(n-1)}{2s(d+1)} + \frac{n(d-1)}{s(d+1)}.$$

Обозначим через $\mathcal{A}_2$ множество всех PIR-протоколов таких, что степень существенности реконструирующей функции не превосходит 2.

Следствие 2. *Если $s \asymp \sqrt{n}$ при $n \to \infty$ и $\log_2 s$ — целое, то при $n \to \infty$ имеет место*

$$C(2,n,s,\mathcal{A}_2) \asymp \sqrt{n}.$$

Автор выражает благодарность Э.Э .Гасанову за постановку задачи.

Список литературы

1. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In Proc. of 36th FOCS, 1998.

О проблеме полноты в функциональной системе линейных полиномов с рациональными коэффициентами

Мамонтов А. И.,
кафедра математического моделирования МЭИ,
140100, Московская обл., г. Раменское, ул. 2-ая Песочная, д. 10.
e-mail: MamontovAI@yandex.ru

В ряде задач возникает универсальная алгебра вида $(P(\mathbb{Q}), C)$, где $P(\mathbb{Q})$ - множество полиномов с рациональными коэффициентами. Исследование этой системы начато в работах [1 – 3]. В нашей работе изучается подалгебра линейных полиномов $L(\mathbb{Q})$.

Из результатов работы [3] вытекает

Теорема 1. *Класс $L(\mathbb{Q})$ является предполным в $P(\mathbb{Q})$.*

Класс $L(\mathbb{Q})$ являвляется бесконечно-порожденным, системы

$$\{1, x-y, \frac{x}{2}, \frac{x}{3}, \ldots, \frac{x}{p}, \ldots\},\ \{1, -x, x+y, \frac{x}{2}, \frac{x}{3}, \ldots, \frac{x}{p}, \ldots\},$$

где p — любое простое число, образуют базисы в $L(\mathbb{Q})$. В настоящей работе мы описываем некоторые предполные классы и формулируем критерий относительной полноты.

Определим следующие замкнутые классы.

1. Пусть $L^+(\mathbb{Q})$ — подкласс всех таких функций из $L(\mathbb{Q})$, у которых все коэффициенты при переменных неотрицательны.

2. Для любого натурального числа n обозначим через $\mathbb{Q}_n$ множество всех несократимых рациональных дробей, знаменатель которых делится на n. Для любого простого p обозначим через $C_p(\mathbb{Q})$ класс всех таких функций из $L(\mathbb{Q})$, у которых все коэффициенты при переменных не принадлежат $\mathbb{Q}_p$.

3. Пусть $E(\mathbb{Q})$ — подкласс всех таких функций из $L(\mathbb{Q})$, у которых сумма коэффициентов при переменных равна 1.

4. Пусть $M(\mathbb{Q})$ — подкласс всех таких функций из $L(\mathbb{Q})$, у которых сумма модулей всех коэффициентов при переменных не больше 1.

5. Пусть $O(\mathbb{Q})$ — подкласс всех функций одной переменной из $L(\mathbb{Q})$.

6. Для любого $a \in \mathbb{Q}$ обозначим через $U_a(\mathbb{Q})$ класс всех функций из $L(\mathbb{Q})$, сохраняющих множество $\{a\}$.

Нами установлено, что эти классы являются предполными в $L(\mathbb{Q})$, и доказана

Теорема 2. *Система функций $G \subseteq L(\mathbb{Q})$, содержащая функцию $f(x,y) = Ax + By + C$, где $A \in \mathbb{Z}$, $|A| > 1$, полна в $L(\mathbb{Q})$ тогда и только тогда, когда при любом простом p и любом $a \in \mathbb{Q}$ эта система не содержится ни в одном из классов $L^+(\mathbb{Q})$, $C_p(\mathbb{Q})$, $E(\mathbb{Q})$, $O(\mathbb{Q})$, $U_a(\mathbb{Q})$.*

Нами найдены бесконечные базисы в каждом из указанных классов.

В [2] установлены следующие результаты:

Теорема 3. *Не существует алоритма, распознающего образует ли произвольная конечная система функций из $P(\mathbb{Q})$ вместе с множеством всех функций одной переменной из $P(\mathbb{Q})$ полную систему в $P(\mathbb{Q})$.*

Теорема 4. *Не существует алоритма, распознающего образует ли произвольная конечная система функций из $P(\mathbb{Q})$ вместе с множеством всех одночленов из $P(\mathbb{Q})$ полную систему в $P(\mathbb{Q})$.*

Нами доказаны

Теорема 5. *Если функция $f \in L(\mathbb{Q}) \setminus O(\mathbb{Q})$, то система $O(\mathbb{Q}) \cup \{f\}$ полна в $L(\mathbb{Q})$.*

Теорема 6. *Если функция $f \in L(\mathbb{Q}) \setminus O(\mathbb{Q})$, то объединение $\{f\}$ и множества всех одночленов из $L(\mathbb{Q})$ образует полную систему в $L(\mathbb{Q})$.*

Пусть $L(\mathbb{N})$ и $L(\mathbb{Z})$ — соответственно классы всех линейных полиномов с целыми неотрицательными и целыми коэффициентами. Аналогично результатам [2] доказывается

Теорема 7. *Глубина замкнутых классов $L(\mathbb{N})$ и $L(\mathbb{Z})$ в замкнутом классе $L(\mathbb{Q})$ равна $\aleph_0$.*

Список литературы

[1] Дарсалия В.Ш. Условия полноты для полиномов с натуральными, целыми и рациональными коэффициентами // Фундаментальная и прикладная математика, 1996. – Т. 2, вып. 2. – С. 365–374.

[2] Дарсалия В.Ш. Проблема полноты для функциональных систем полиномов // Дисс. канд. физ.-мат. наук, Москва, 1998.

[3] Горбань А.Н. Обобщенная апроксимационная теорема и точное представление многочленов от нескольких переменных суперпозициями многочленов от одного переменного // Известия вузов. Математика, 1998. вып. 5. – С. 6–9.

Синтез оптимального прибора в задачах нелинейной редукции измерения

О. В. Мондрус, К. С. Соболев
119992, ГСП-2, Москва, Ленинские горы,
МГУ им.М.В.Ломоносова, Физический факультет
e-mail: olya@cmp.phys.msu.su, sob@cmp.phys.msu.su

Введениие. В настоящее время хорошо разработаны линейные методы редукции измерения на основе теории измерительно-вычислительных систем (ИВС), [1]. Для класса задач линейного программирования существуют и методы синтеза линейных измерительных приборов на ИВС. Также разработаны методы и решен класс задач теоретико-возможностной редукции измерения, но синтез прибора на нелинейных моделях пока что оставался неосвещенным.

В данном докладе приводится результат решения задачи интервальной редукции измерения, синтез оптимального измерительного прибора $\overset{\circ}{U}$ на ИВС интервальной модели $[A, I_f, I_\nu]$ (интервальной редукции измерения) (см.[1],[2]), а также теорема, резюмирующая полученные результаты для синтеза измерительного прибора на ИВС интервальных моделей нелинейной редукции измерения.

Напомним понятие редукции измерения на примере идеализированного варианта - задачи несмещенной редукции измерения. Предположим, что задана модель $[A, \Sigma]$ схемы измерения

$$\xi = Af + \nu, \tag{1}$$

где $A : \mathcal{R}_m \to \mathcal{R}_n$ — оператор математической модели измерительного прибора, $\Sigma : \mathcal{R}_n \to \mathcal{R}_n$ — корреляционный оператор шума ν, моделирующего погрешность измерения, f — априори произвольный вектор $\mathcal{R}_m$, $\mathbf{E}\nu = 0$. В задаче несмещенной редукции предполагается заданным линейный оператор $U : \mathcal{R}_m \to \mathcal{R}_k$, символизирующий измерительный прибор, выходной сигнал которого желательно синтезировать на ИВС, и требуется определить линейный оператор $R : \mathcal{R}_n \to \mathcal{R}_k$ так, чтобы $R\xi$ можно было интепретировать как наиболее точную версию Uf, каким бы ни был входнй сигнал $f \in \mathcal{R}_m$.

Ошибку редукции будем определять как $\sup\limits_{f\in\mathcal{R}_m} ||R\xi - Uf||^2$. Оператор R определим из условия: $E_f||R\xi - Uf||^2 = \mathrm{trR\Sigma R^*} \sim \min\limits_{\mathrm{R}}$, где min вычисляется на множестве операторов, удовлетворяющих условию $RA = U$. Если корреляционный оператор Σ шума ν не вырожден, а оператор U удовлетворяет условию $U(I - A^-A) = 0 \Leftrightarrow RA = U$, то $R\xi = U(A^*\Sigma^{-1}A)^-A^*\Sigma^{-1}\xi = U(\Sigma^{-1/2}A)^-\Sigma^{-1/2}\xi$ — искомое наилучшее приближение Uf, $f \in \mathcal{R}_m$, а $R\xi = Uf + U(\Sigma^{-1/2}A)^-\Sigma^{-1/2}\nu$ — оценка сопутствующей погрешности оценивания.

Пусть $\overset{\circ}{U}$ — прибор, который требуется синтезировать на ИВС $[A, \Sigma]$, $\mathcal{U}_\delta$ — класс матриц $n \times n$ вида:

$$U_\delta(\alpha_i) = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \dots & \alpha_n & 0 & 0 & 0 & 0 & \dots \\ \alpha_1 & \alpha_0 & \alpha_1 & \alpha_2 & \dots & \alpha_n & 0 & 0 & 0 & \dots \\ \alpha_2 & \alpha_1 & \alpha_0 & \alpha_1 & \alpha_2 & \dots & \alpha_n & 0 & 0 & \dots \\ \dots & & & & & & & & & \end{pmatrix}, \tag{2}$$

где α_i, $i = 1, \dots, 2\delta + 1$ — элементы матрицы оператора U, которые могут принимать поизвольные значения.

Так как по сути $\mathcal{U}_\delta$ - класс приборов приемлемого качества, то задачу синтеза можно поставить как задачу отыскания прибора $U_\varepsilon \in \mathcal{U}$ и наиближайшего к нему прибора $R_\varepsilon A$, такого, что

$$||U_\varepsilon - R_\varepsilon A||_2^2 = \inf\{||U - RA||_2^2 | U \in \mathcal{U}, R, \mathbf{E}||R\nu||^2 \leqslant \varepsilon\}. \tag{3}$$

Класс $\mathcal{U} = \mathcal{U}_\delta$ содержит идеальный прибор $\overset{\circ}{U} = U_0 = I$ и все те приборы, у которых аппаратная функция «не шире», чем $2\delta + 1$, то есть моделирует приборы, разрешение которых не хуже $2\delta + 1$.

Теорема 1. *Пусть* $\Sigma > 0$. $U_\delta(\omega) = E_0 + \sum\limits_{\lambda=1}^{\delta} a_\lambda(\omega)E_\lambda$ — *решение системы уравнений*

$$\omega\frac{\partial}{\partial\lambda}\mathrm{tr(US^{-1}(\omega)U^*)} = 0,\ \lambda = 1, \dots, \delta,\ \textit{где}\ \mathrm{S(\omega) = A^*\Sigma^{-1}A + \omega I}.$$

Решение задачи (3) *имеет вид:*

$$R_{\varepsilon,\delta} = \begin{cases} U_\delta(\omega)S^{-1}(\omega)A^*\Sigma^{-1}, \omega = \omega(\varepsilon,\delta), & 0 < \varepsilon < \varepsilon(\delta), \\ U_\delta(+0)(\Sigma^{-1/2}A)^-\Sigma^{-1/2}, & \varepsilon \geqslant \varepsilon(\delta), \\ 0, & \varepsilon = 0; \end{cases}$$

$$U_{\varepsilon,\delta} = \begin{cases} U_\delta(\omega), \omega = \omega(\varepsilon,\delta), & 0 < \varepsilon < \varepsilon(\delta), \\ U_\delta(+0), & \varepsilon \geqslant \varepsilon(\delta), \\ U_\delta(+\infty), & \varepsilon = 0, \end{cases}$$

где $U_\delta(+0) = \lim\limits_{\varepsilon\to+0} U_\delta(\omega)$, $U_\delta(+\infty) = \lim\limits_{\varepsilon\to\infty} U_\delta(\omega)$, $\varepsilon(\delta) = \mathrm{tr}\mathrm{U}_\delta(+0)(\mathrm{A}\Sigma^{-1}\mathrm{A})^-\mathrm{U}_\delta^*(+0)$ *и* $\omega(\varepsilon,\delta)$ *— единственный корень уравнения* $\mathrm{tr}\mathrm{U}_\delta(\omega)\mathrm{A}^*\Sigma^{-1}\mathrm{AS}^{-2}\mathrm{U}_\delta(\omega) = \varepsilon$.

Данную задачу можно интерпретировать как задачу несмещенного синтеза прибора приемлемого качества с минимальным уровнем шума. Возможны и другие постановки, как правило многокритериальные. Подробнее о синтезе измерительного прибора для задач линейной редукции измерения см. [1].

Интервальная редукция измерения. Представим схему измерения (1) в виде

$$\xi_i = a_{i1}f_1 + \ldots + a_{im}f_m + \nu_i \quad i = 1,\ldots n, \tag{4}$$

в которой $\xi \triangleq \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}$ — результат измерений вектора $f \triangleq \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix}$, $\nu \triangleq \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_n \end{pmatrix}$ — вектор погрешностей измерений, причем известно, что в (4) $\nu \in I_\nu \triangleq [\underline{\nu}, \overline{\nu}] \triangleq \begin{pmatrix} I_{\nu_1} \\ \vdots \\ I_{\nu_n} \end{pmatrix}$, где $I_{\nu_i} = [\underline{\nu}_i, \overline{\nu}_i]$, $i = 1,\ldots,n$, и $f \in I_f \triangleq [\underline{f}, \overline{f}] \triangleq \begin{pmatrix} I_{f_1} \\ \vdots \\ I_{f_m} \end{pmatrix}$, где $I_{f_j} = [\underline{f}_j, \overline{f}_j]$, $j = 1,\ldots,m$, $A \triangleq \begin{pmatrix} a_{11}\ldots a_{1m} \\ \vdots \\ a_{n1}\ldots a_{nm} \end{pmatrix}$ — известная матрица оператора, моделирующего измерительный прибор.

Таким образом, задана интервальная модель $[A, I_f, I_\nu]$ схемы измерений (4), [1]. Согласно равенствам (4) для любых значений $f_1,\ldots,f_m$ и учитывая, что любой интервал $\underline{f} \leqslant f \leqslant \overline{f}$ может быть представлен в виде $I_j \sim \{c_j, l_j\}$, $j = 1,\ldots,m$, где c_j —центр интервала I_j, l_j — полудлина интервала I_j, то

$$\xi_i - \underline{\nu}_i \geqslant \sum_{j=1}^m a_{ij}c_j + \sum_{j=1}^m |a_{ij}|l_j \geqslant \sum_{j=1}^m a_{ij}c_j - \sum_{j=1}^m |a_{ij}|l_j \geqslant \xi_i - \overline{\nu}_i, \tag{5}$$

$$\underline{f}_j \leqslant c_j - l_j \leqslant c_j + l_j \leqslant \overline{f}_j, j = 1,\ldots,n. \tag{6}$$

В задаче редукции измерения (4) задана матрица

$$U = \begin{pmatrix} u_{11}\ldots u_{1m} \\ \vdots \\ u_{n1}\ldots u_{nm} \end{pmatrix}$$

оператора, моделирующего идеальный измерительный прибор, и требуется наиболее точно оценить его отклик

$$Uf = u \triangleq \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} \sum\limits_{j=1}^m u_{1j}f_j \\ \vdots \\ \sum\limits_{j=1}^m u_{nj}f_j \end{pmatrix}$$

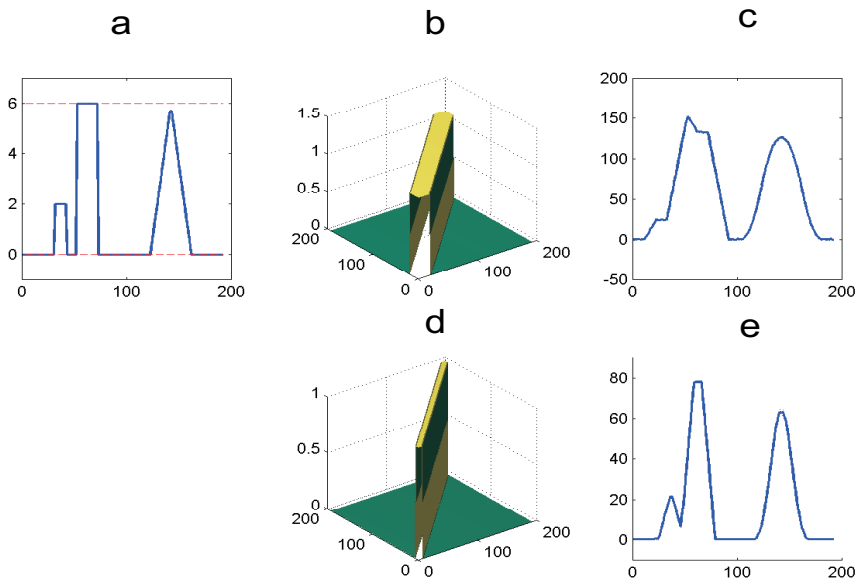


Рис. 1: а) Входной сигнал $f \in \mathcal{R}_m$ ИВС; б) график матрицы A ($\mathcal{R}_m \to \mathcal{R}_n$) оператора, моделирующего измерительный прибор; в) выходной сигнал $\xi \in \mathcal{R}_n$ ИВС; г) график матрицы U ($\mathcal{R}_m \to \mathcal{R}_n$) оператора, представляющего идеальный измерительный прибор; д) решение задачи редукции измерения к прибору U (7).

на входной сигнал f, измеренный в (4) с помощью прибора A. Речь идет о задаче редукции измерения (4), к виду, свойственному измерению, выполненному на идеальном приборе U, [1,2].

Задача редукции измерения может быть сформулирована как задача на максимум

$$\sum_{i=1}^{n}\sum_{j=1}^{m}|u_{ij}|l_j \sim \max_{(c,l)\in D(A,I_f,I_\nu|\xi)} \tag{7}$$

при ограничениях (5), (6) ($D(A, I_f, I_\nu|\xi)$),

Решением задачи редукции измерения является пара $\{c^*, l^*\}$, где c^* — оценка вектора f, l^* — погрешность редукции (7) измерения (4). $\{Uc^*, Ul^*\}$ - результат редукции (7), представленный на рис. 1.

Синтез измерительного прибора. Рассмотрим задачи синтеза оптимального измерительного прибора на ИВС для интервальной модели $[A, I_f, I_\nu]$ нелинейной редукции измерения, рассмотренной выше.

Для начала определим класс $\mathcal{U}$ приборов U как класс матриц $n \times n$ вида (2). Для каждого набора значений $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ на классе $\mathcal{U}$ решим задачу:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}|u_{ij}|l_j \sim \min_{U\in\mathcal{U}} \max_{(c,l)\in D(A,I_f,I_\nu|\xi)}, \tag{8}$$

результатом решения которой будет нахождение погрешности, минимальной по $U \in \mathcal{U}$, тем самым мы определим оптимальный прибор U с минимальной погрешностью редукции.

Рассмотрим условия (6), исключив неравенства $\underline{f}_j \leqslant c_j - l_j \leqslant c_j + l_j \leqslant \overline{f}$, $j = 1, \ldots, m$.

$$D_l = \begin{pmatrix} \sum\limits_{j=1}^{m}|a_{ij}|l_j \leqslant \xi_i - \underline{\nu}_i - q_i, \\ \sum\limits_{j=1}^{m}|a_{ij}|l_j \leqslant \overline{\nu}_i + q_i - \xi_i, \\ i = 1, \ldots, m, \end{pmatrix}, \; q_i = \sum_{j=1}^{m} a_{ij}c_j, i = 1, \ldots, m. \tag{9}$$

Так как речь идет о задаче (8), в которой максимум вычисляется на множестве $D_l = D(A, I_\nu|\xi)$,

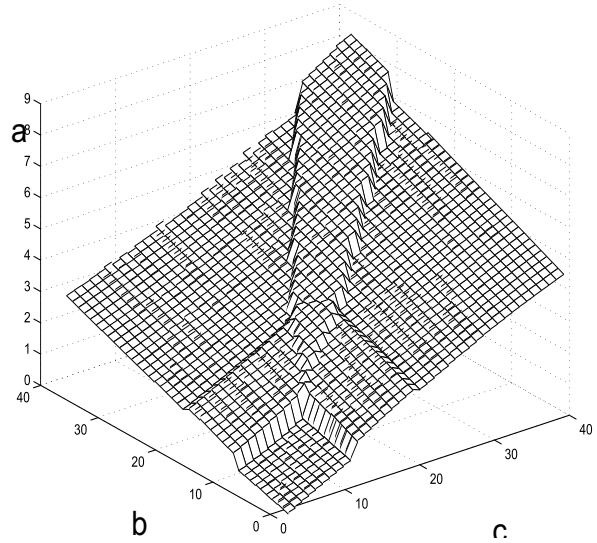


Рис. 2: Решение задачи (11) для случая $i = 1, 2$. По горизонтальным осям: значения диагональных элементов α_i оператора U, $0 \leqslant \alpha_i \leqslant 1.5$, по вертикальной оси - погрешность σ редукции (12) измерения (4).

$q_1, \ldots, q_m$ в (9) определяются из условий:

$$\min(\xi_i - \underline{\nu}_i - q_i, \underline{\nu}_i + q_i - \xi_i) \sim \max_{q_i}, i = 1, \ldots, m. \tag{10}$$

Максимум достигается при q_i, удовлетворяющем условию: $\xi_i - \underline{\nu}_i - q_i = \underline{\nu}_i + q_i - \xi_i$ $i = 1, \ldots, m$, т.е. при

$$q_i = q_i^*(\xi) = \xi_i - \frac{\overline{\nu}_i - \underline{\nu}_i}{2}, i = 1, \ldots, m.$$

Матрица $\{a_{ij}\}$ в (10) — невырожденная, отсюда заключаем, что существуют единственные $c_0^*(\xi) = \begin{pmatrix} c_1^* \\ \vdots \\ c_m^* \end{pmatrix}$, при которых $c_j = c_j^*(\xi) = \sum_{k=1}^{m} a_{jk}^- q_k^*(\xi), j = 1, \ldots, m$, где $\{a_{jk}^-\}$ — матрица, обратная $\{a_{kj}\}$. Условия (9) представим в виде $D_l = \sum_{j=1}^{m} |a_{ij}| l_j \leqslant \frac{\overline{\nu}_i - \underline{\nu}_i}{2}, i = 1, \ldots, m$. Решив задачу

$$\sum_{i=1}^{n} \sum_{j=1}^{m} |u_{ij}| l_j \sim \min_{U \in \mathcal{U}} \max_{l \in D_l}, \tag{11}$$

получим решение $\{c_0^*(\xi), l_0^*\}$ задачи редукции измерения для модели $[A, I_\nu|\xi]$

$$l_0^* = \min_{U \in \mathcal{U}} l \ (l = \max_{l \in D_l} \sum_{i=1}^{n} \sum_{j=1}^{m} |u_{ij}| l_j), \tag{12}$$

эквивалентное решению задачи (8).

Представим задачу (11), следующим образом:

$$F(\alpha, l) \triangleq ||U(\alpha_i) l|| = \sum_{i=1}^{m} \sum_{j=1}^{m} |u_{ij}| l_j = \sum_{j=1}^{m} (1 + \sum_{k=1}^{n} \eta_{jk} \alpha_k) l_j, \tag{13}$$

где η_{jk} — целые положительные числа, $l_j \in D_l$, см. (9). Множество $D_l \in \mathcal{R}_m$ выпукло и ограничено, и при любом фиксированном α существует $\max_{l \in D_l} F(\alpha, l) = F(\alpha, l(\alpha)) = G(\alpha)$.

Проведенные исследования позволяют сформулировать теорему для общего случая поиска оптимального измерительного прибора U ИВС $[A, I_\nu|\xi]$:

Теорема 2.

$$\min_{\alpha \in \mathcal{R}_m} G(\alpha) = G(o), \ o = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{14}$$

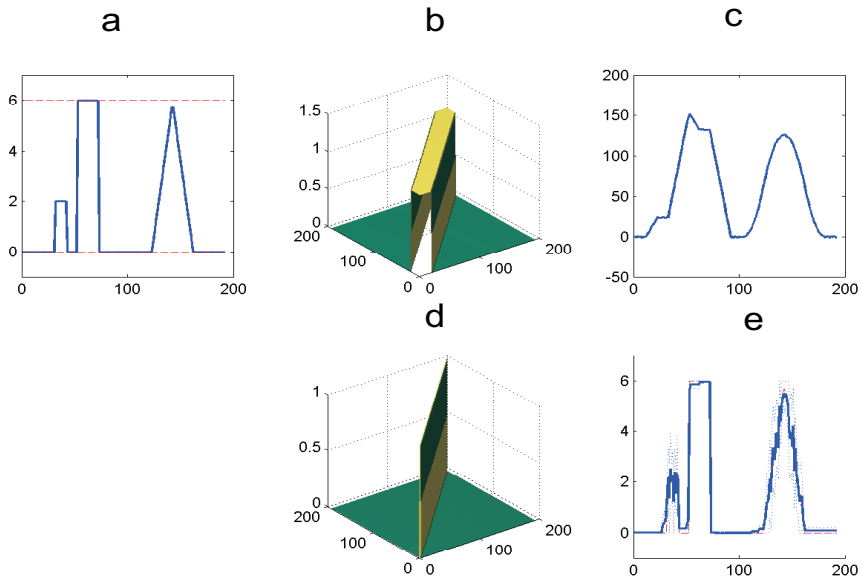


Рис. 3: а) Входной сигнал $f \in \mathcal{R}_m$ ИВС; б) график матрицы A ($\mathcal{R}_m \to \mathcal{R}_n$) оператора, моделирующего измерительный прибор; в) выходной сигнал $\xi \in \mathcal{R}_n$ ИВС; г) график матрицы U ($\mathcal{R}_m \to \mathcal{R}_n$) оператора, представляющего идеальный измерительный прибор; д) решение задачи редукции измерения к оптимальному измерительному прибору $U = I$ (7)

т.е. минимум функционала $G(o)$ достигается тогда и только тогда, когда $U = \overset{\circ}{U} = I$.

Доказательство. При любом фиксированном $l \in D_l$, (9), минимум $F(\alpha, l) = \sum_{j=1}^{m}(1 + \sum_{k=1}^{n}\eta_{jk}\alpha_k)l_j$ достигается при $\alpha_j = 0$, т.к. $l_j \geqslant 0$ и $\eta_{kj} \geqslant 0$, т.е.

$$\min_{\alpha \in \mathcal{R}_m} F(\alpha, l) = F(o, l) = \sum_{j=1}^{m} l_j. \tag{15}$$

Пусть $\omega_0 = \max_{l \in D_l} F(0, l_0) = F(o, l_0^*)$, тогда

$$\begin{array}{l} G(\alpha) = \max_{l \in D_l} F(\alpha, l) \geqslant F(\alpha, l^*) \geqslant F(o, l_0^*) = \omega_0, \\ \text{т.е.} \quad \min_{\alpha \in \mathcal{R}_m} \max_{l \in D_l} F(\alpha, l) \geqslant F(0, l_0^*). \end{array} \tag{16}$$

□

Итак, в то время, как для статистического случая, изложенного во введении, в результате синтеза прибора, мы приходим к нетривиальному решению $U \neq I$, в теоретико-возможностной интервальной постановке мы приходим к единственно возможному тривиальному решению $\overset{\circ}{U} = I$ на классе приборов $\mathcal{U}$. Иллюстрация решения задачи (11) для случая подбора двух диагоналей оператора U ИВС $[A, I_\nu|\xi]$ приведена на рис.2. На рис.3. приведено решение задачи редукции измерения (7) к подобранному идеальному прибору $\overset{\circ}{U}$, (8), (11).

Список литературы

1. *Ю. П. Пытьев.* Методы математического моделирования измерительно-вычислительных систем, М.: Физиматлит, 2004

2. *Ю. П. Пытьев.* Возможность как альтернатива вероятности. Математические и эмпирические основы применения М.: Физматлит 2007

3. *Ю. П. Пытьев.* Математические методы интерпретации эксперимента. - М.:Высшая школа, 1989. - 352 с.

Параллельный доступ к данным без раскрытия запроса

Назаров М. Н., *E-mail: nazmax@mail.ru*
к.ф.-м.н., доцент кафедры информационных систем и математического моделирования
Волгоградской академии государственной службы, 400033, г. Волгоград, ул. Менжинского, д.15, кв.128.

Рассмотрим задачу обращения пользователя к базе данных для получения необходимой информации при условии, что сервера, на которых хранится база данных, не должны узнать адрес запрошенной информации. Более точно база данных это булев вектор $x = (x_1, x_2,..., x_n)$. Пользователь запрашивает i-й бит данного вектора, желая сохранить номер бита i в секрете от серверов, на которых хранится данный булев вектор.

Такая задача была рассмотрены в [1] и получила название Private Information Retrieval, а протоколы, решающие ее, называются PIR-протоколами или протоколами доступа к данным без раскрытия запроса [2]. Как было отмечено в [1] и [2] такие протоколы должны удовлетворять условию защищенности. А именно, независимо от содержания вектора x, для любых двух индексов i и j ни один из серверов, получивших запрос, не должен определить, какой именно бит интересует пользователя. Суммарное количество бит, участвующих в обмене информацией между пользователем и базой данных называется коммуникационной сложностью PIR-протокола. Ясно, что тривиальным решением задачи является получение пользователем всей базы данных с последующим выбором нужного бита. То есть в худшем случае коммуникационная сложность протокола равна длине базы данных.

Определим теперь базу данных, как таблицу для булевой функции n переменных $f(x_1, x_2,..., x_n)$. Каждый набор значений переменных является адресом, по которому хранится один бит информации. В такой постановке задача доступа к данным без раскрытия запроса заключается в вычислении данной булевой функции на заданном наборе n переменных $(x_1, x_2,..., x_n)$ при условии, что набор переменных останется неизвестным серверам базы данных.

Рассмотрим протокол с более мягким условием защищенности. Пусть является допустимым, что каждый сервер может получить из запроса пользователя часть адреса, который интересует пользователя, но не весь адрес. Например, пусть пользователя интересует информация по адресу $X = (x_1, x_2,..., x_n)$. Представим набор X в виде $X=AB$, где $A= (x_1, x_2,..., x_k)$ и $B = (x_{k+1}, x_{k+2},..., x_n)$. Тогда в результате запроса допускается, что каждый сервер базы данных может получить часть адреса B, но часть A остается секретной. Назовем такие протоколы частичными PIR-протоколами.

В работе [3] был предложен параллельный алгоритм вычисления булевой функции n переменных на r независимых наборах при условии, что допускается только одно обращение к базе данных, содержащей данную булеву функцию. Алгоритм получил название $DP_{r,f}$.

Заметим, что для работы алгоритма $DP_{r,f}$ необходимо разбить исходное пространство $\{0,1\}^n$ на два подпространства $\{0,1\}^n = \{0,1\}^{4k} \times \{0,1\}^{n-4k}$, где $k = \lceil 3\log r + \log n \rceil$. Тогда все входы $X_1, X_2, \ldots, X_r$ можно представить в виде $X_i = A_i B_i$, $i=1, 2, \ldots, r$, где $A_i \in \{0,1\}^{4k}$, а $B_i \in \{0,1\}^{n-4k}$. Исходная функция $f(x_1, x_2,..., x_n)$ реализуется через вспомогательные булевы функции f_A и $g_l : \{0,1\}^{n-4k} \to \{0,1\}$, где $k = \lceil 3\log r + \log n \rceil$. Для большей наглядности разбиение булевой функции f можно представить в виде таблицы для булевых функций от $n - 4k$ переменных.

Таблица 1

y_1	y_2	...	y_{n-4k}	f_{A_1}	f_{A_2}	...	f_{A_m}	g_1	g_2	...	g_p
0	0	...	0	*	*	...	*	*	*	...	*
0	0	...	1	*	*	...	*	*	*	...	*
...	...	...	...	...	...	...	...	...	...	...	...
1	1	...	1	*	*	...	*	*	*	...	*

Здесь $m = 2^{4k}$, а $p = 2^{4k}/n$. Каждый столбец данной таблицы хранится на отдельном носителе. Допуск одного обращения означает, что для любых r входов из каждого столбца таблицы можно извлечь только один бит.

Как было показано в [4] при работе алгоритма считывается по одному биту из каждого столбца таблицы 1, используя только r строк, соответствующих наборам B_i. Следовательно, данный алгоритм можно использовать в качестве частичного PIR-протокола, организующего параллельный доступ к r битам информации без раскрытия запросов.

Действительно, каждый сервер, хранящий один столбец таблицы 1, получает запрос, соответствующий какому-то набору B_i, i=1, 2, . . . , r. Никакой информации о первой части адресов запросов ни один сервер не получает. Пользователь восстанавливает необходимую ему информацию, используя реконструирующую функцию. Коммуникационная сложность такого частичного PIR-протокола определяется количеством столбцов таблицы 1 и не превосходит величину

$$(2^{4k} + 2^{4k}/n) \le 16r^{12}n^4(1+1/n).$$

Откажемся теперь от допуска частичного раскрытия адреса запросов и рассмотрим PIR-протокол, построенный на алгоритме $DP_{r,f}$.

Пусть необходимо получить бит из базы данных по адресу X_1, то есть вычислить функцию базы данных $f(X_1)$. Выберем случайным образом произвольный набор X_2 и отправим запрос к базе данных, позволяющий получить сразу два значения $f(X_1)$ и $f(X_2)$. Применим алгоритм $DP_{2,f}$ для получения нужной информации. В соответствии с данным алгоритмом из каждой части разбиения базы данных в таблице 1 будет считан ровно один бит. При этом каждый сервер не знает, считывает ли он бит по адресу B_1, являющимся частью секретного адреса пользователя X_1 или по адресу B_2, который является частью адреса, выбранного случайным образом. Таким образом, полученный протокол удовлетворяет условию защищенности и является протоколом без раскрытия запроса.

Оценим сложность такого PIR-протокола. Коммуникационная сложность определяется величиной

$$16 \cdot 2^{12}n^4(1+1/n) = 65536n^4(1+1/n).$$

Отметим, что такая сложность не является слишком большой при больших базах данных, так как их размерность определяется величиной 2^n. Тем более, что избыточность информации по сравнению с исходной базой данных составляет $2^n/n$, в то время, как обычные PIR-протоколы требуют как минимум копирования всей базы данных на два носителя.

Однако, можно получить более хорошие оценки коммуникационной сложности. Применим аналогичную схему запросов при использовании в качестве PIR-протокола алгоритма $MDP_{2,f}$, построенного в [4], для параллельного вычисления булевой функции на двух независимых наборах. В этом случае информация считывается не со всех частей разбиения базы данных, но при этом каждый сервер по прежнему не знает, считывает ли он информацию по адресу, необходимому пользователю или по случайному адресу. Кроме того, как следует из [4] оба бита, соответствующие $f(X_1)$ и $f(X_2)$ вообще не считываются, а соответствующие им значения вычисляются. Коммуникационная сложность такого PIR-протокола определяется как 32n, так как количество считываемых бит из базы данных для алгоритма $MDP_{r,f}$ определяется величиной $2r^4n$.

К другим оценкам сложности можно отнести время параллельной работы алгоритма $MDP_{r,f}$, размер базы данных и количество процессоров, необходимых для его работы. Время работы алгоритма оценивается величиной

$$512n^3\log(r^3n)\log(\log(r^3n)) + \textbf{extime},$$

где величина **extime** оценивает время обращения к внешнему носителю информации. Размерность базы данных вместе с добавочной информацией составляет $2^n(1+1/n)$ бит. Для параллельной работы алгоритма необходимо 32n процессоров.

Список литературы.

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval //In Proc. of the 36th Annu. IEEE Symp. on Foundations of Computer Science, pages 41–51, 1995. Journal version: J. of the ACM, 45:965–981, 1998.

[2] Гасанов Э. Э., Майлыбаева Г. А. Доступ к базам данных без раскрытия запросов //Материалы конференции «Математика и безопасные информационные технологии», Москва, 23-24 октября 2003 г. – 2003. – С. 393-395.

[3] Andreev A. E., Clementi A. E. F., Rolim J. D. P. On the Parallel Computations of Boolean Functions on Unrelated Inputs //IV IEEE Israel Symposium on Theory of Computing and Systems (ISTCS'96). — IEEE. — 1996. — P.155—161

[4] Назаров М. Н. Параллельный доступ к базам данных и булевы функции //Интеллектуальные системы. – 2003. – Т. 7. – С. 365-381.

Асимптотика сложности разбиения булевого куба на подкубы

Осокин В. В., *e-mail: mail@osvic.ru*
кафедра математической теории интеллектуальных систем МГУ им. М.В.Ломоносова

Рассмотрим k-мерный булев куб B^k. Известно, что любой $(k-s)$-мерный подкуб B^k можно задавать некоторой конъюнкцией $x_{i_1}^{\sigma_1}\dots x_{i_s}^{\sigma_s}$, $s \in \{1,\dots,k\}$, $\sigma_i \in \{0,1\}$, $i \in \{1,\dots,s\}$. *Тематическим разбиением* R k-мерного булевого куба B^k назовем произвольное упорядоченное множество подкубов этого куба такое, что

а) пересечение любых двух элементов R пусто;

б) объединение всех элементов R дает B^k;

в) для любой переменной x_j, $j \in \{1,\dots,k\}$ существует такой элемент множества R, что переменная x_j входит в соответствующую этому элементу элементарную конъюнкцию.

Элементы тематического разбиения R будем называть темами.

Пусть k, n — натуральные числа, $k \leqslant n$, R — тематическое разбиение k-мерного куба, $\{y_1,\dots,y_n\}$ — множество булевых переменных. Функцию $f(y_1,\dots,y_n) : B^n \to \{1,2,\dots,|R|\}$ будем называть *темоопределяющей* над разбиением R k-мерного булевого куба, если существует упорядоченное множество индексов $q_1,\dots,q_k$, такое что

а) $q_j \in \{1,2,\dots,n\}$ и $q_{j_1} \neq q_{j_2}$, если $j_1 \neq j_2$, $j, j_1, j_2 \in \{1,\dots,k\}$;

б) для произвольной j-й темы ($j \in \{1,2,\dots,|R|\}$) $x_{i_1}^{\sigma_1}\dots x_{i_{t_j}}^{\sigma_{t_j}}$ разбиения R из того, что $y_{q_{i_1}} = \sigma_1, \dots, y_{q_{i_{t_j}}} = \sigma_{t_j}$ следует, что $f(y_1,\dots,y_n) = j$;

при этом множество индексов $\{q_1,\dots,q_k\}$ будем называть *основанием* темоопределяющей функции f, а переменные $y_{q_1},\dots,y_{q_k}$ — *тематическими переменными* функции f. Обозначим $y_{q_1} = \Psi(x_1),\dots,y_{q_k} = \Psi(x_k)$.

Через Φ_R^n обозначим множество в точности всех темоопределяющих функций от n переменных над разбиением R. Пусть задано некоторое тематическое разбиение R и задан оператор A_f, вычисляющий для произвольного набора из B^n значение некоторой функции $f \in \Phi_R^n$ на этом наборе . Задача данной работы — за минимальное число обращений к оператору A_f полностью восстановить таблицу значений функции $f(y_1,\dots,y_n)$.

Рассмотрим множество $\mathcal{F}$ алгоритмов, решающих указанную задачу для любого разбиения $R \in \mathcal{R}_k$ и для любой функции $f \in \Phi_R^n$, где $\mathcal{R}_k$ — множество всех тематических разбиений куба B^k. На вход любого алгоритма F из этого множества подаются разбиение $R \in \mathcal{R}_k$ и оператор $\mathcal{A}_f$. На выходе F выдает таблицу значений функции f, или, что то же самое, номера k искомых тематических переменных. Другими словами, при фиксированном тематическом разбиении R для произвольной $f \in \Phi_R^n$ любой алгоритм $F \in \mathcal{F}$ с помощью оператора A_f полностью восстанавливает таблицу значений функции f. Работа алгоритма F заключается в том, что он последовательно запрашивает значения оператора $\mathcal{A}_f$ на наборах из B^n. При этом алгоритм F предполагается условным, т.е. при выборе очередного набора он может пользоваться знаниями о тематических переменных, полученными на ранее поданных им наборах. Любой тройке — алгоритму $F \in \mathcal{F}$, тематическому разбиению $R \in \mathcal{R}_k$ и функции $f \in \Phi_R^n$ — можно сопоставить число $\varphi(F,R,n,f)$ — число обращений к оператору A_f в процессе восстановления таблицы значений функции f с помощью алгоритма F. Обозначим $\varphi(k,n) = \min_{F\in\mathcal{F}} \max_{R\in\mathcal{R}_k} \max_{f\in\Phi_R^n} \varphi(F,R,n,f)$.

Теорема 1. *Если $k \to \infty$ при $n \to \infty$ и $k \leqslant cn$, где $c < 1$, то при $n \to \infty$ $\varphi(k,n) \sim k \log_2 n$.*

Покажем, что $\varphi(1,n) \geqslant]\log_2 n[$. Будем считать в одномерном случае, что тема $\overline{x}_1$ имеет номер 0, а тема x_1 имеет номер 1. Фиксируем произвольный алгоритм G_0. Шагом алгоритма G_0 будем считать запрос значения оператора $\mathcal{A}_f$ на некотором наборе. Пусть a_i — набор, сгенерированный алгоритмом G_0 на i-м шаге, a_i^j — j-я компонента этого набора, $i \in \{1,2,\dots\}$, $j \in \{1,\dots,n\}$. На каждом i-м шаге будем строить множество $T_i = \{y_{j_1},\dots,y_{j_i}\}$, $j_i \in \{1,\dots,n\}$, переменных, которые все еще могут быть тематической переменной. Для любых $j \in \{1,\dots,n\}$, $i \in \{1,2,\dots\}$ переменная y_j лежит в множестве T_i тогда и только тогда, когда, $\mathcal{A}_f(a_s) = a_s^j$ для всех $s \in \{1,\dots,i\}$. Положим для удобства $T_0 = \{y_1,\dots,y_n\}$. Очевидно, $T_0 \supseteq T_1 \supseteq T_2 \supseteq \dots$. Очевидно также, что алгоритм G_0 решает задачу нахождения тематической переменной на s-м шаге тогда и только тогда, когда $|T_s| = 1$. Пусть $a_i' = (a_i^{j_1},\dots,a_i^{j_{i-1}})$, где $T_{i-1} = \{y_{j_1},\dots,y_{j_{i-1}}\}$. Наконец, положим для любого $i \in \{1,2,\dots\}$ $z(a_i) = 0$, если в наборе a_i' нулей больше, чем единиц, $z(a_i) = 1$ в противном случае.

Построим для алгоритма G_0 «плохую» темоопределяющую функцию f, для определения которой алгоритм запросит значения оператора $\mathcal{A}_f$ минимум на $]\log_2 n[$ наборах. Заметим, что до тех пор, пока мы не определили тематическую переменную, значение оператора на произвольном наборе мы можем выбирать произвольным образом. На каждом i-м шаге, $i \in \{1,2,\dots\}$, полагаем $\mathcal{A}_f(a_i) = z(a_i)$. При указанном способе выбора значений оператора $\mathcal{A}_f$ множеству T_i,

$i \in \{1, 2, \ldots\}$, принадлежит не менее половины переменных множества T_{i-1}. Значит, потребуется минимум $s =]\log_2 n[$ наборов для того, чтобы мощность множества T_s была единицей. Отсюда $\varphi(1, n) \geqslant]\log_2 n[$.

Перейдем к k-мерному случаю. В нем разбиений уже много, как и темоопределяющих функций, удовлетворяющих этим разбиениям. Поэтому при выборе самой плохой ситуации у нас есть возможность варьировать и разбиение, и функцию.

Лемма 1. *Имеет место* $\varphi(k, n) \geqslant (k -]\log_2 k[)]\log_2(n - k + 1)[$.

Идея доказательства состоит том, что существует такое тематическое разбиение R куба, задаваемого переменными $x_1, \ldots, x_k$, что никакая пара переменных множества $\{x_{]\log_2 k[+1}, \ldots, x_k\}$ не встречается в одной и той же теме разбиения R. Тогда по аналогии с одномерным случаем можно построить такую «плохую» для заданного алгоритма функцию, что на определение каждой переменной из $\{x_{]\log_2 k[+1}, \ldots, x_k\}$ потребуется не менее $]\log_2(n - k + 1)[$ запросов значений темоопределяющей функции. Отсюда можно показать, что всего алгоритм сгенерирует не менее $(k -]\log_2 k[)]\log_2(n - k + 1)[$ наборов.

Перейдем к построению верхней оценки. Снова начнем с одномерного случая. В одномерном случае существует единственное возможное тематическое разбиение R одномерного булева куба: $x_1 \vee \overline{x}_1$ и n возможных темоопределяющих функций, подходящих под это разбиение (т.е. $|\Phi_R^n| = n$). Построим алгоритм, определяющий любую из этих n функций за $]\log_2 n[$ запросов значений соответствующего оператора.

Упорядоченное множество наборов A из B^n назовем *правильным*, если в матрице, строками которой являются эти наборы, все n столбцов различны. Эту матрицу также будем обозначать A. Очевидно, существует правильное множество мощности $]\log_2 n[$, т.к. существует $2^{]\log_2 n[} \geqslant n$ различных столбцов длины $]\log_2 n[$. Через F_A обозначим алгоритм, который последовательно запрашивает значения оператора $\mathcal{A}_f$ на всех строках $\{a_1, \ldots, a_{|A|}\}$ матрицы A.

Если R — тематическое разбиение одномерного куба, то для любой функции $f \in \Phi_R^n$ для любого правильного множества A мощности $]\log_2 n[$ выполнено $\varphi(F_A, R, n, f) \leqslant]\log_2 n[$. Действительно, пусть $c = (c_1, \ldots, c_{]\log_2 n[})$, где $c_s = 0$, если $\mathcal{A}_f(a_s) = \mathcal{N}(\overline{x}_1)$, $c_s = 1$, если $\mathcal{A}_f(a_s) = \mathcal{N}(x_1)$, $s \in \{1, \ldots,]\log_2 n[\}$. Так как искомая тематическая переменная существует, то в матрице, образованной наборами множества A, найдется столбец $b_l = c$, $l \in \{1, \ldots, n\}$. По определению правильного множества такой столбец единственен. Легко видеть, что переменная y_l и является искомой тематической. Значит, алгоритм F_A действительно решает задачу определения одной тематической переменной.

В силу произвольности f из неравенства $\varphi(F_A, R, n, f) \leqslant]\log_2 n[$ следует неравенство $\varphi(1, n) \leqslant]\log_2 n[$. Итак, для одномерного случая имеем точную оценку $\varphi(1, n) =]\log_2 n[$.

Опишем алгоритм F_0, такой что для любых $R \in \mathcal{R}_k$, $f \in \Phi_R^n$ выполнено $\varphi(F_0, R, n, f) \leqslant 2k]\log_2 n[+ k$. Рассмотрим некоторую матрицу A. Считаем, что ее строки и столбцы можно помечать, пусть m_1 — массив номеров помеченных строк, m_2 — массив номеров помеченных столбцов. Через $\mathcal{J} = (\mathcal{J}_1, \ldots, \mathcal{J}_k)$ обозначим вектор длины k, такой что $\mathcal{J}_j = l$, если $\Psi(x_j) = y_l, \mathcal{J}_j = *$, если значение $\Psi(x_j)$ неизвестно, $j \in \{1, \ldots, k\}$, $l \in \{1, \ldots, n\}$. Вектор $\mathcal{J}$ назовем *ответом*. Будем называть переменную x_j, $j \in \{1, \ldots, k\}$ *найденной*, если $\mathcal{J}_j \neq *$.

В основе алгоритма F_0 лежит функция $RECURS(A, R, m_1, m_2)$. Опишем ее работу. Последовательно вычисляем значения оператора $\mathcal{A}_f$ на непомеченных строках матрицы A (пусть это строки $a_1, \ldots, a_d$). Возможны 2 случая.

1. Мы находим строку a_i, $i \in \{a_1, \ldots, a_d\}$: $\mathcal{A}_f(a_i)$ — некрайняя тема. Назовем a_i разделяющим набором, а $\mathcal{A}_f(a_i)$ — разделяющей темой, пусть эта тема задается конъюнкцией $x_{i_1} \ldots x_{i_r}\overline{x}_{i_{r+1}} \ldots \overline{x}_{i_t}$, $t \in \{1, \ldots, k\}$. Рассмотрим все столбцы $b_{i_1}, \ldots, b_{i_s}$, $s \in \{1, \ldots, n\}$, матрицы A, пересечения которых со строкой a_i — нули. Определим новую матрицу $A' = (b'_1, \ldots, b'_n)$, где для $j \in \{1, \ldots, n\}$ $b'_j = b_j$, если $j \neq i_1, \ldots, i_s$, b'_j — нулевой столбец в противном случае. Пометим строку a'_i, положив $m'_1 = m_1 \cup \{i\}$ и столбцы $b'_{i_1}, \ldots, b'_{i_s}$, положив $m'_2 = m_2 \cup \{i_1, \ldots, i_s\}$. Рассмотрим разбиение R', полученное из разбиения R фиксацией переменных $x_{i_{r+1}}, \ldots, x_{i_t}$ нулями. Вызываем функцию $RECURS(A', R', m'_1, m'_2)$. Теперь рассмотрим все столбцы $b_{l_1}, \ldots, b_{l_h}$, $h \in \{1, \ldots, n\}$, матрицы A, пересечения которых со строкой a_i — единицы. Определим новую матрицу $A'' = (b''_1, \ldots, b''_n)$, где для $j \in \{1, \ldots, n\}$ $b''_j = b_j$, если $j \neq l_1, \ldots, l_h$, b''_j — единичный столбец в противном случае. Пометим строку a''_i, положив $m''_1 = m_1 \cup \{i\}$ и столбцы $b''_{l_1}, \ldots, b''_{l_h}$, положив $m''_2 = m_2 \cup \{l_1, \ldots, l_h\}$. Рассмотрим разбиение R'', полученное из разбиения R фиксацией переменных $x_{i_1}, \ldots, x_{i_r}$ единицами. Полученное разбиение обозначим R''. Вызываем функцию $RECURS(A'', R'', m''_1, m''_2)$.

2. Для любого $i \in \{1, \ldots, d\}$ $\mathcal{A}_f(a_i)$ — одна из двух крайних тем. Считаем, что среди непомеченных строк матрицы A есть и лежащие в отрицательной крайней теме, и в положительной. Иначе заменяем все элементы строки a_1, соответствующие непомеченным столбцам, их от-

рицаниями (запросив одно дополнительное значение оператора $\mathcal{A}_f$). При этом строка a_1 уже не обязана лежать в крайней теме. Берем первую переменную x_j, $j \in \{1,\dots,k\}$, которая входит во все темы $\mathcal{A}_f(a_i)$, $i \in \{1,\dots,d\}$. Если такой переменной не существует, то выдаем ошибку и заканчиваем работу. Пусть $c = (c_1,\dots,c_d)$, где $c_s = 0$, если в $\mathcal{A}_f(a_s)$ x_j входит с отрицанием, $c_s = 1$, если в $\mathcal{A}_f(a_s)$ x_j входит без отрицания, $s \in \{1,\dots,d\}$. Среди непомеченных столбцов матрицы, образованной строками $a_1,\dots,a_d$ находим столбец $b_l = c$, $l \in \{1,\dots,n\}$. Переменная y_l является тематической. Помещаем ее в ответ, полагая $\mathcal{J}_j = l$.

1. Устанавливаем $\mathcal{J} = (*,\dots,*)$, $R' = R$, $n' = n$. Пусть A — матрица $]\log_2 n[\times n$, соответствующая произвольному правильному множеству. Полагаем $A' = A$.

2. Полагаем $m_1 = \oslash$, $m_2 = \oslash$ и вызываем функцию $RECURS(A', R', m_1, m_2)$.

3. Если в векторе ответа $\mathcal{J}$ все компоненты не равны $*$, т.е. определились, то алгоритм F_0 завершает работу.

4. Фиксируем найденные переменные таким образом, чтобы разбиение R', полученное из R операцией фиксации этих переменных, зависело хотя бы от одной переменной. Такая фиксация обязательно существует.

5. Пусть $Q = \{q_{i_1},\dots,q_{i_l}\}$, $l < k$, — множество номеров всех найденных тематических переменных. Положим $n' = n - l$. Фиксируем произвольное правильное множество с матрицей A' размера $]\log_2 n'[\times n'$. Нумеруем ее столбцы числами из множества $\{1,\dots,n\}\backslash Q$. Пусть матрица A'' получена из матрицы A' добавлением l столбцов, имеющих в матрице A'' номера $q_{i_1},\dots,q_{i_l}$. Причем столбец матрицы A'' с номером q_{i_p}, $p \in \{1,\dots,l\}$ — единичный, если переменная x_{i_p} фиксирована единицей, и нулевой, если x_{i_p} фиксирована нулем. Под значением оператора $\mathcal{A}_f$ на некоторой строке матрицы A' будем понимать значение $\mathcal{A}_f$ на строке матрицы A'' с тем же номером.

6. Идем на шаг 2.

Приведем идею построения упомянутой оценки.

Каждой паре $R \in \mathcal{R}_k$, $f \in \Phi_R^n$, подаваемой на вход алгоритма F_0, алгоритм сопоставляет некоторое число бинарных деревьев $\{\mathcal{D}_1,\dots,\mathcal{D}_s\}$. Общее количество концевых вершин всех деревьев равно k, в каждой из этих k вершин F_0 запрашивает значения функции f на не более чем $]\log_2 n[+1$ наборах. В каждой неконцевой вершине запрашиваются значения f на не более чем $]\log_2 n[$ наборах. Пусть в дереве $\mathcal{D}_i$, $i \in \{1,\dots,s\}$ k_i концевых вершин. Учитывая, что в любом бинарном дереве количество неконцевых вершин не превышает количество концевых, получаем, что в вершинах дерева D_i алгоритм F_0 запрашивает не более $2k_i]\log_2 n[+k_i$ значений функции f. Тогда всего может понадобиться не более $\sum_{i=1}^{s}(2k_i]\log_2 n[+k_i) = 2k]\log_2 n[+k$ значений f. Отсюда для любых $R \in \mathcal{R}_k$, $n > k$, $f \in \Phi_R^n$ имеет место $\varphi(F_0, R, n, f) \leqslant 2k]\log_2 n[+k$.

На базе F_0 строится алгоритм F_1 (здесь алгоритм F_1 не приводится), решающий задачу определения k тематических переменных, для которого $\varphi(F_1, R, n, f) \leqslant k]\log_2 n[+2k$, откуда получаем следующую лемму.

Лемма 2. *Имеет место* $\varphi(k, n) \leqslant k]\log_2 n[+2k$.

Согласно леммам 1 и 2 $(k-]\log_2 k[)]\log_2(n-k+1)[\leqslant \varphi(k,n) \leqslant k]\log_2 n[+2k$, из чего следует утверждение теоремы.

Автор выражает благодарность профессору Э.Э.Гасанову за постановку задачи и помощь в работе.

Об отличимости автоматов при искажениях на входе

Пантелеев П. А.,

Москва, Воробьевы горы, МГУ, механико-математический факультет

E-mail: panteleev@intsys.msu.ru

Работа посвящена изучению надежной отличимости состояний конечных автоматов при искажении информации на входе автомата.

Рассмотрим конечный автомат $\mathfrak{A} = (A, Q, B, \varphi, \psi)$, где A, Q, B — *входной алфавит*, *алфавит состояний* и *выходной алфавит*, а $\varphi : Q \times A \to Q$ и $\psi : Q \times A \to B$ — *функции переходов* и *выходов*, соответственно. Назовем два его состояния q_1, q_2 *k-кратно отличимыми* словом α, если они отличимы любым словом α', которое отличается от α в не более чем k позициях. Два состояния q_1, q_2 *k-кратно отличимы*, если существует слово, которое их k-кратно отличает. Если такого слова нет, то q_1, q_2 называются *k-кратно неотличимыми*. Очевидно, что 0-кратная отличимость совпадает с обычной отличимостью состояний. Обозначим через $\mathcal{K}_n$ класс всех автоматов с n состояниями.

Пусть для состояний q_1, q_2 автомата $\mathfrak{A}$ существует k-кратно отличающее слово. Обозначим через $l^k(\mathfrak{A}, q_1, q_2)$ длину минимального такого слова и 0 если его не существует. Рассмотрим следующую

функцию Шеннона:

$$L^k(\mathcal{K}_n) = \max_{\mathfrak{A}\in\mathcal{K}_n, q_1, q_2} l^k(\mathfrak{A}, q_1, q_2),$$

где максимум берется по всем автоматам $\mathfrak{A} \in \mathcal{K}_n$ и парам q_1, q_2 их состояний.

В общем случае задача оказалась достаточно трудной и пока удалось получить некоторые оценки логарифма величины $L^k(\mathcal{K}_n)$.

Теорема 1. $n \lesssim \log_2 L^k(\mathcal{K}_n) \lesssim kn^2/2$ *при* $n \to \infty$, $k > 0$.

Если ограничиться случаем $k = 1$, то справедлива более точная верхняя оценка.

Теорема 2. *Имеет место* $\ln L^1(\mathcal{K}_n) < n \ln n$ *при* $n > 1$.

Автомат, использованный для получения нижней оценки в теореме 1, имеет экспоненциальную от числа состояний мощность входного алфавита. Однако даже если ограничится классом $\mathcal{K}_{2,n,2}$ автоматов с n состояниями и двухбуквенным входным и выходным алфавитом, то нижняя оценка по-прежнему экспоненциальна от n.

Теорема 3. *Существует такой автомат* $\mathfrak{A} \in \mathcal{K}_{2,n,2}$ *и два состояния* q_1, q_2 *в нем, что* $l^k(\mathfrak{A}, q_1, q_2) \gtrsim \sqrt{n \ln n}$ *при* $n \to \infty$.

Определение. Два состояния q_1, q_2 автомата $\mathfrak{A}$ называются *ω-кратно отличимыми*, если они k-кратно отличимы для любого $k \geqslant 0$.

Теорема 4. *Если два состояния* q_1, q_2 *автомата* $\mathfrak{A} \in \mathcal{K}_n$ *k-кратно отличимы, где* $k \geqslant \frac{n(n-1)}{2}$, *то они ω-кратно отличимы.*

Теорема 5. *Для каждого целого* n, $n \geqslant 2$, *существует автомат* $\mathfrak{A} \in \mathcal{K}_n$ *такой, что для любого* $k \in \{0, 1, \ldots, \frac{n(n-1)}{2} - 1\}$ *найдется пара состояний* q_1, q_2, *которая k-кратно отличима, но не* $(k+1)$*-кратно отличима.*

При проектировании автомата мы обычно хотим быть уверены, что у него нет состояний неотличимых с точки зрения внешнего поведения. На этом пути возникло понятие приведенного автомата. Если при этом на входе автомата могут происходит искажения, то мало потребовать его приведенности. Разумно потребовать, чтобы все его состояния были попарно ω-отличимы, т.е. при любом числе k искажений на входе у автомата не будет k-кратно неотличимых состояний.

Определение. Автомат называется *кратно-приведенным*, если любая его пара отличимых состояний 1-кратно отличима.

Как показывает следующая теорема класс $\mathcal{C}$ кратно-приведенных автоматов в точности удовлетворяет нашим требованиям.

Теорема 6. *Если автомат* $\mathfrak{A}$ *кратно-приведенный, то все его состояния ω-кратно отличимы.*

Следствие 1. *Если для автомата* $\mathfrak{A} \in K_n$ *существует слово, k-кратно отличающее все пары его различных состояний и* $k > 0$, *то всегда найдется такое слово длины не более* $(k+1)\frac{n(n-1)}{2}$.

Обозначим через $\mathcal{C}_n$ — множество кратно-приведенных автоматов с n состояниями. Рассмотрим для этого класса функцию Шеннона $L^k(\mathcal{C}_n)$ k-кратной отличимости двух состояний.

Теорема 7. $L^k(\mathcal{C}_n) = (n - 1 + k)n(n-1)/2$.

Автор выражает глубокую и искреннюю благодарность академику Валерию Борисовичу Кудрявцеву и профессору Александру Сергеевичу Подколзину за постановку задач, постоянную поддержку и внимание к работе.

Список литературы.

[1.] Moore E. F. Gedanken-experiments on sequential machines Automata Studies, 1956, p. 129-153[русский перевод см. "Автоматы"(сб. статей), 1956, ИЛ, с. 179-210]

[2.] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Элементы теории автоматов Изд-во МГУ, 1985, 320 с

[3.] Пантелеев П. А. Об отличимости состояний автоматов Дискретная математика, 2003, т. 15, вып. 3, с. 76-90

[4.] Пантелеев П. А. Об отличимости состояний решетчатых автоматов Интеллектуальные системы, 2005, т. 8, с. 529-542

[5.] Panteleev P. A. On distinguishability of states of automata Discrete Mathematics and Applications, 2003, vol. 13, num. 4, p. 355-370

Методы нелинейной динамики в моделировании эволюции солнечной активности[1]

Перепелица В. А., *e-mail: perepel2@yandex.ru*
профессор кафедры компьютерной безопасности
Ставропольского государственного университета;
369015, г.Черкесск-15, а/я 32;
Тебуева Ф. Б., *e-mail: fariza-t@yandex.ru*
доцент кафедры прикладной математики Карачаево-Черкесской
государственной технологической академии;
369000, г. Черкесск, ул. Ставропольская, д.36;

Используемые в настоящей работе термины, понятия и факты, относящиеся к пятнообразовательной деятельности Солнца, читатель может найти в [1]. Временные ряды (ВР) чисел Вольфа представлены на сайте [2]. В контексте предпрогнозного анализа и прогнозирования временных рядов чисел Вольфа (среднемесячных и среднегодичных) авторы настоящей работы предлагают использовать такой метод нлинейно динамики [3], как инструментарий фазового анализа [4], который позволяет выявлять новые закономерности, не обнаруживаемые с помощью математической статистики.

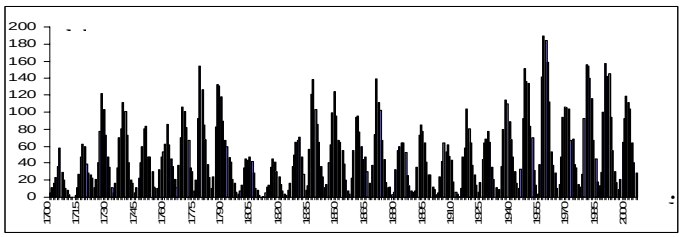


Рисунок 1. Графическое представление временного ряда среднегодичной солнечной активности за период 1700-2005 гг.

ВР (среднегодичных) значений чисел Вольфа обозначим через $W=\langle w_i \rangle$, где индексом $i=1,2,...,n$ занумерованы годы с 1700 по 2005. В целях визуализации на рис.1 дано графическое представление ВР W. Используемый авторами фазовый анализ базируется на построении фазовой траектории $\Phi_2(W)=\{(w_i, w_{i+1})\}$, $i=\overline{1,n-1}$, представленной на рис.2, где пары соседних точек (w_i, w_{i+1}), (w_{i+1}, w_{i+2}) соединяются отрезком кривой. Эта фазовая траектория разбивается на фазовые квазициклы K_r, $r=\overline{1,28}$. Число точек в квазицикле K_r называется его длиной и обозначается через L_r. В качестве типичного на рис.3 представлен квазицикл K_{26}, у которого длина L_{26}, равная 10, означает, что он является 11- летним, т.е. состоит из 11 уровней ВР W. Представленный на рис.1 ВР W фактически состоит из 28 завершенных квазициклов, которые в совокупности включают в себя 306 (среднегодичных) уровней W. Отсюда получаем среднее значение длины квазициклов вида рис.3 $L_{cp}=306/28\approx 10{,}93$ лет, т.е. оказалось вычисленным среднее значение длительности так называемого «11-летнего цикла» [1].

На рис.4 дано графическое представление частот длин квазициклов ВР W, откуда вытекает также 11-летнее значение средней длины годичных квазициклов; в области значений длин $\{9,10,....,14\}$ типичными являются 10 и 11 лет. Отметим, что представленные на рис.4 данные в терминологии [1] определены согласно правилам «эпохи минимумов» (эти данные в [1] определяют продолжительность цикла в пределах от 9,0 до 13,6).

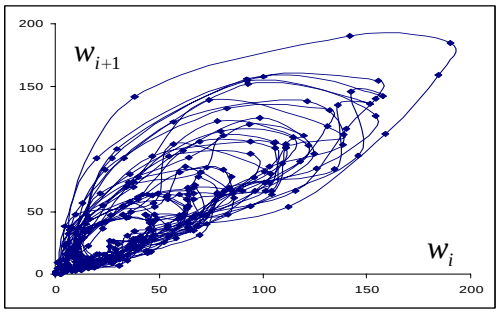


Рисунок 2. Фазовая траектория временного ряда среднегодичной солнечной активности W

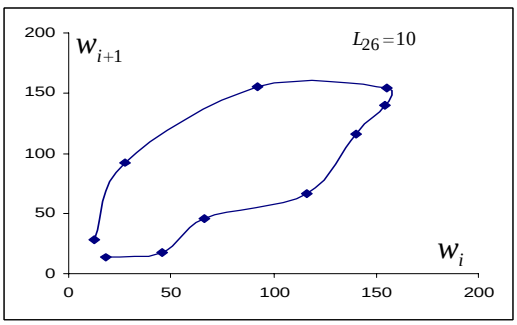


Рисунок 3. Типичный квазицикл с длиной $L_{26}=10$ в фазовой траектории временного ряда среднегодичной солнечной активности W

[1] Работа выполнена при поддержке гранта РФФИ, проект № 06-01-00020а

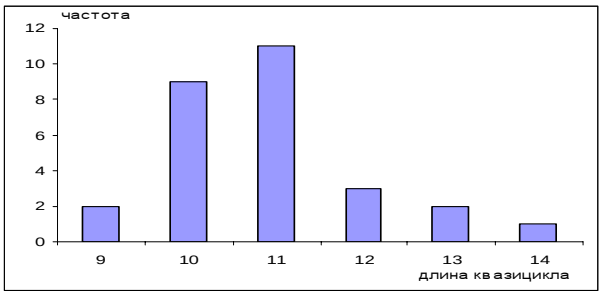
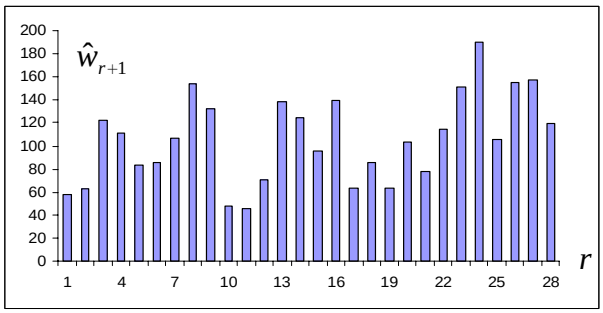
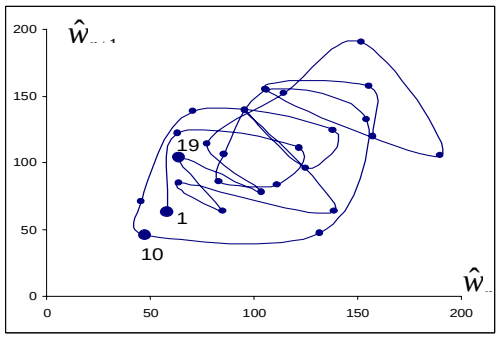


Рисунок 4. Распределение частот длин квазициклов временного ряда среднегодичной солнечной активности W

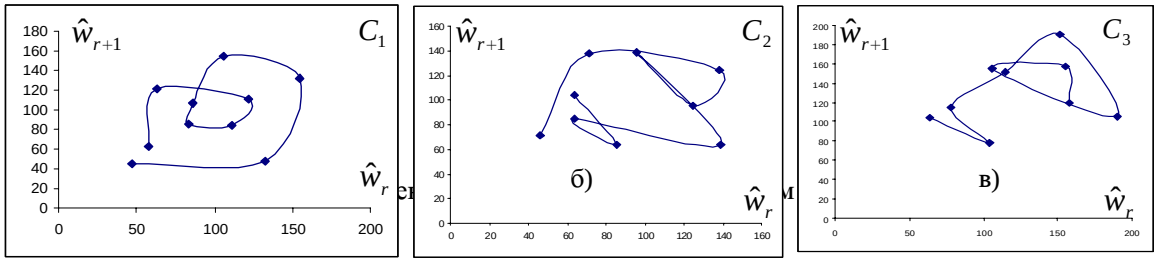
Остановимся теперь на вопросе (“вероятно существующего” [1]) векового цикла пятнообразования. Представленная в [1] библиография позволяет назвать следующие 3 подхода в попытках различных авторов обосновать существование векового цикла и оценить его продолжительность: 1) метод векового сглаживания (В. Глайзберг, Д. Шов, М. Эйгенсон, М. Вальдмайер), 2) метод скользящих средних (Б. Рубашев, Ю. Витинский), а также использование спектрального анализа и индексов мощности явлений пятнообразования. Различные подходы привели к различным оценкам продолжительности вековых циклов: 79 лет, 80-90 лет и др.





Описанию предлагаемого ниже метода фазовых траекторий предпошлем одно замечание. В [1] отмечено следующее предложение В.Ф. Чистякова: «вековой цикл начинается и заканчивается максимумом». Фазовый анализ выделения векового квазицикла базируется на ВР локальных максимумов $\hat{W} = \langle \hat{w}_r \rangle$, $r = \overline{1,m}$, где $\hat{w}_r$ – это значение максимального уровня в r-ом квазицикле ВР W, $m = 28$ – число наблюдаемых квазициклов. На рис.5 дано графическое представление ВР $\hat{W}$, а на рис.6 приведена фазовая траектория $\Phi_2(\hat{W}) = \{(\hat{w}_r, \hat{w}_{r+1})\}$, $r = \overline{1,m-1}$. На рис.7 а,б,в представлено разложение этой фазовой траектории на 3 сложных квазицикла C_1, C_2 и C_3 (термин «сложный» означает наличие внутри квазицикла петли, происхождение которой объяснил В.Ф. Чистяков: «вековой цикл обычно характеризуется двумя максимумами, которые разделены «провалом»). В терминах нумерации точек фазовой траектории $\Phi_2(W)$ на рис.2 цикл C_1 начинается в точке 6 и заканчивается в точке 105, т.е. его длина $L(C_1)$ равна 105-6+1=100 лет; цикл C_2 начинается в точке 105 и заканчивается в точке 206, т.е. его длина $L(C_2)$ равна 206-105+1=102 года; тогда длина цикла C_3 составляет $L(C_3) = 300 - 206 + 1 = 95$ лет. Оставляя пока в стороне вопрос о том, является ли завершенным или незавершенным цикл C_3 с учетом значений $L(C_1) = 100$, $L(C_2) = 102$, условимся называть вековым пересекающиеся циклы C_1, C_2, C_3 (локальные максимумы $\hat{w}_{10}$ и $\hat{w}_{19}$ выполняют двоякую роль – конец одного цикла и начало другого).

Рисунок 7. Вековые циклы временного ряда среднегодичной солнечной активности W



- в фазовой траектории $\Phi_2(\hat{w})$ точки начала этих квазициклов сосредоточены в узкой окрестности (на рис.6 см. точки с номерами 1,10,19);

- каждый из квазициклов длится порядка одного столетия в земных годах;

- в терминах фазовой траектории локальных максимумов $\Phi_2(\hat{w})$ на рис.6 структура каждого квазицикла удовлетворяет правилу «восход – петля (упомянутая выше) – нисход»;

- в измерении количества 11-летних циклов для вековых C_1, C_2 и C_3 выполняются соотношения 0,5+8+0,5, где первое (третье) слагаемое означает ветвь спада (роста) начального (конечного) 11-летнего цикла в составе векового цикла.

Из этих четырех особенностей заключительная, возможно, не является обязательной с учетом того, что у третьего векового цикла его длина $L(C_3)=95$ земных лет и, кроме того, на рис.7в точка окончания цикла C_3 находится на значительном расстоянии от точки его начала. Последнее дает некоторые основания предположить, что следующий 11-ти летний цикл K_r, $r=29$ окажется достаточно коротким и незначительным по величине локального максимума $\hat{w}_{29}$. В этом случае присоединение этого (пока не существующего) максимума к точке завершения векового квазицикла C_3 на рис.7в окажется достаточно близкой к точке его начала. При этом в единицах измерения земного года вековой цикл C_3 может оказаться на несколько лет длиннее C_2.

Рассмотрим результаты фазового анализа ВР среднемесячных чисел Вольфа $\tilde{W}=\langle\tilde{w}_j\rangle$, $j=\overline{1,k}$. В качестве иллюстративного примера рассмотрим период с января 1981 г по декабрь 2005 г., т.е. $k=300$. Фазовая траектория ВР среднемесячных чисел Вольфа $\Phi_2(\tilde{W})=\{(\tilde{w}_j,\tilde{w}_{j+1})\}$, $j=\overline{1,k}$ в определенном смысле содержит циклическую компоненту, которая, однако, проявляет принципиальное отличие от циклической компоненты фазовой траектории ВР среднегодичных чисел Вольфа (см. рисунки 2, 3 и 4). На рис.8 дано графическое представление распределения частостей длин квазициклов ВР среднемесячной солнечной активности $\tilde{W}$. Здесь поведение квазициклов длины 3 принципиально отличается от квазициклов длины $l\geq 4$. По существу в случае $l=3$ соответствующие части фазовой траектории не являются квазициклами, а представляют лишь последовательности ацикличных отрезков, т.е. зигзагообразных отрезков фазовой траектории. Эти ацикличные отрезки появляются в окрестности смены знака приращений («+» на «-» или «-» на «+») среднегодичного ВР W (см. рис.1), т.е. в окрестности локальных точек максимума (или минимума) этого ВР. Собственно «циклическая часть» фазовой траектории $\Phi_2(\tilde{W})$ состоит из квазициклов, которые имеют в среднем полугодичную длительность.

Резюмируя итоги настоящего исследования, можем отметить их совпадение или согласованность с общепринятыми фактами [1] в той части, которая касается 11-летних циклов среднегодичного ВР солнечной активности.

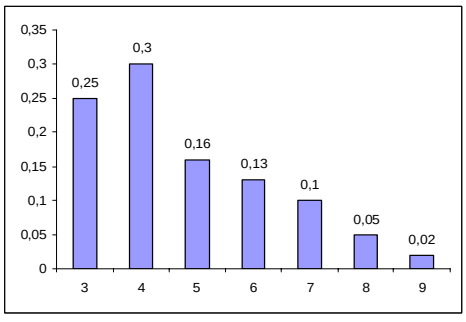


Рисунок 8. Распределение частостей длин квазициклов временного ряда среднемесячной солнечной активности $\tilde{W}$

которая касается вековых циклов, а также квазициклов среднемесячного ВР солнечной активности. Заслуживает внимания также тот факт, что фазовый анализ ВР солнечной активности устанавливает трехуровневую иерархию а структуре динамики эволюционирования уровней солнечной активности. Нижний уровень этой иерархии составляют квазициклы среднемесячного ВР, средний уровень составляют 11-летниеквазициклы среднегодичного ВР чисел Вольфа, верхний уровень составляют вековые циклы. Более подробный анализ этих уровней оперирует такими параметрами, как траектория дрейфа центров квазициклов, траектория размера периметров их габаритных прямоугольников и др.

Список литературы

1. Витинский Ю. И., Копецкий М., Куклин Г. В. Статистика пятнообразовательной деятельности Солнца. – М.: Наука, 1986. – 296 с.

2. R. Van der Linden/ http://sidc.oma.be

3. Малинецкий Г. Г. Математические основы синергетики. Хаос, структуры, вычислительный эксперимент. – М.: КомКнига, 2005. – 312 с.

4. Перепелица В. А., Тебуева Ф. Б., Савина Л. А. Формирование предпрогнозной информации методами фазового анализа для временных рядов я памятью/ Труды 1-го Международного форума

«Актуальные проблемы современной науки». Естественные науки. – Части 1,2: Математика. Математическое моделирование. – Самара: Изд-во Сам ГТУ, 2005. – С.76-79.

О модели асимптотической оценки ресурсоемкости реляционного запроса

Плашенков Валерий Владимирович, *д.в.н., профессор*,
Борчук Леонид Евгеньевич, *E-mail:le_borchuk@mail.ru*
Череповецкий государственный университет
162600, Череповец, ул. Наседкина 9-24.

В настоящее время для настройки запросов пользователей к реляционной системе управления базами данных используется метод экспертных оценок. Предлагаемая ниже математическая модель основана на теории асимптотических оценок количества действий и представляет собой попытку математического моделирования запроса. Модель отличается от известных независимостью от физической реализации реляционной системы и достаточной простотой, что позволяет моделировать процесс настройки запросов пользователей в условиях статистических оценок малого количества измеряемых параметров.

Ключевые слова: Реляционный запрос, Реляционное отношение, Реляционная операция, Асимптотическая оценка, Испытания Бернулли, Минимальный граф зависимостей, Пространство итераций, Алгебра А, Эквивалентность реляционных отношений.

1. Введение

Современные системы управления реляционными базами данных (РСУБД) позволяют получить требуемую информацию, не вдаваясь в подробности физической реализации и способы обработки данных. Однако, вопреки ожиданиям, по мере роста объемов хранимых данных, описание требуемой информации приходится уточнять либо изменять с целью получить более эффективный способ обработки.

Способы (операции) обработки информации, хранимой в виде кортежей реляционных отношений, фиксированы системой выполнения запроса. Один и тот же ответ на запрос пользователя может быть получен различными комбинациями операций за счет свойств коммутативности, транзитивности, ассоциативности и других известных свойств операций. Задача выбора способа выполнения запроса состоит в оптимизации затрат ресурсов в соответствии с критерием оптимальности для различных комбинаций операций (составляющих пространство поиска). Возможные комбинации операций определяются запросом пользователя, представляющим описание информационных зависимостей данных, хранимых в РСУБД.

Для пользователя время ответа системы не должно превышать некоторую заданную величину (как правило, от 2 до 30 секунд). Данному показателю могут отвечать несколько способов выполнения запроса. Описание информационных зависимостей на языке запросов при этом может быть недостаточно точным (полным). Достаточно, чтобы пространство поиска включало в себя хотя бы одну удовлетворяющую показателю точку. С ростом объемов данных количество отвечающих показателю точек уменьшается. В связи с этим требуется улучшить полноту описания информационных зависимостей, либо, когда описать более полно уже нет возможности, увеличить количество операций обработки путем создания индексов или материализованных представлений.

В связи с ростом объемов хранимой информации количество запросов, требующих настройки, постоянно возрастает. Предлагаемая ниже математическая модель на основе асимптотических оценок представляет собой попытку математического моделирования влияния параметров участвующих в получении результата реляционных отношений на время ответа системы.

2. Общая постановка задачи

Объектом исследования является способ выполнения реляционного запроса пользователя. Выражением “реляционный запрос” будем обозначать тот факт, что отношения нормализованы, т.е. находятся в первой нормальной форме, и результат любой реляционной операции над реляционными отношениями - реляционное отношение.

Время выполнения запроса $T_з$ будем определять на основании значений затрат ресурсов по формуле:

$$T_з = T_{ож.диск}(D_{диск}) + T_{ож.память}(D_{память}) + T_{обработки}(D_{данные}), \quad (1)$$

где $T_{ож. диск}$ – время ожидания чтения требуемых блоков данных с диска;

$D_{диск}$ – объем данных, считываемых диска;
$T_{ож.\ память}$ – время ожидания извлечения данных из буфера памяти;
$D_{память}$ – объем данных, извлекаемых из памяти;
$T_{обработки}$ – время, затраченное на обработку извлеченных данных;
$D_{данные}$ – объем данных;

Так что задача определения времени выполнения запроса сводится к задаче определения затрат ресурсов.

Предметом исследования являются затраты ресурсов системы управления базой данных на выполнение реляционного запроса пользователя.

Целью построения математической модели является сокращение сроков настройки реляционного запроса путем применения аналитических методов математического анализа. При этом модель должна отвечать требованиям адекватности, достаточной простоты и аддитивности. Исходя из этих требований, учитывая неизвестность алгоритма физической реализации реляционных операций, в качестве вида модели была выбрана модель на основе асимптотических оценок количества действий алгоритма.

Компонентами математической модели являются шаги плана выполнения или реляционные операции над отношениями. Эндогенными переменными математической модели являются затраты ресурсов на выполнение реляционной операции. Экзогенными переменными являются вероятности использования ресурсов при обработке одного кортежа реляционного отношения и статистическая количественная информация о хранимых в РСУБД отношениях и выполняемых над ними реляционных операциях (компонентах). Значения экзогенных и эндогенных переменных конечны.

3. Асимптотическая математическая модель реляционного запроса

Пусть задано конечное множество типов K={k_i, i=1,2,…,M}, для каждого $k_i \in K$ задано счетное множество значений типов V_i, называемое доменом.

Кортежем называется конечная последовательность объектов, возможно заданная в виде списка конечных номеров объектов. n-мерным кортежем называется кортеж, содержащий n объектов. Определим понятие реляционного кортежа u. Его особенностями являются невозможность наличия двух и более объектов одного типа и независимость от порядка следования объектов. Дополним множества V_i “пустым” элементом □, $V_i' = V_i \cup \{\square\}$. Выражение u=(.,□,.), $\square \in V_i'$ будет означать, что кортеж u не содержит ни одного элемента множества V_i. Реляционным отношением называется элемент декартова произведения $u = (a_1, a_2, ..., a_M) \in V_1' \times V_2' \times ... \times V_M'$. Множество всех возможных кортежей обозначим U={u, $u \in V_1' \times V_2' \times ... \times V_M'$}. Введем на u функцию ar(u) как количество непустых элементов кортежа. Значение функции ar(u) называется арностью кортежа. Введем на $u = (a_1, a_2, ..., a_M)'$ функцию nu(u) как упорядоченную последовательность номеров позиций, содержащих непустые элементы, nu(u) = (i_1, i_2, …, $i_{ar(u)}$): $a_{i1} \in V_{i1}, a_{i2} \in V_{i2}, ..., a_{iM} \in V_{iM}$, $1 \le i_1 < i_2 < ... < i_{ar(u)} \le M$. Значение функции nu(u) называется заголовком кортежа. Пронумеруем значения функции nu(u)=j, j=1,2,…, $C_M^{ar(u)}$ как номер сочетания ar(u) элементов из M возможных. Введем на множестве U отношение эквивалентности μ, полагая элементы из U эквивалентными при совпадении их арности и заголовка. Смежные классы эквивалентности μ обозначим как $U_{nu(u)}^{ar(u)}$. Реляционным отношением q арности s заголовком j называется подмножество кортежей смежного класса эквивалентности μ, $q \subseteq U_j^s$. Множество всех возможных отношений обозначим $Q = \{q : q \subseteq U_j^s, s = 1,..,M, j = 1,..,C_M^s\}$.

Определим Ω_F как множество бинарных операций F вида : Q×Q→Q. $\mathbf{Q}$=<Q,Ω_F> - алгебра реляционных отношений. Зададим множество P (P⊂Q), представляющее собой множество отношений, хранящихся в БД в материализованном виде (на носителе). Из множества P может быть получено множество

Q_1={q: q=F(p_1, p_2), $\forall p_1, p_2 \in P, \forall F \in \Omega_F$};
Q_2={q: q=F(q_1, q_2), $\forall q_1, q_2 \in P \cup Q_1, \forall F \in \Omega_F$};
…
Q_i={q: q=F(q_1, q_2), $\forall q_1, q_2 \in \left(\bigcup_{j=1}^{i-1} Q_j\right) \cup P, \forall F \in \Omega_F$}; … $Q = \left(\bigcup_{j=1}^{\infty} Q_j\right) \cup P$.

Возможность получить множество всех возможных отношений Q из множества P показал Э. Ф. Кодд [1].

Таким образом, заданная в виде **Q** алгебра реляционных отношений имеет P порождающим множеством. Множество Q\P представляет собой множество производных отношений БД, вычисляемых по необходимости с затратами определенных вычислительных ресурсов.

Введем на множестве Q отношение эквивалентности σ такое, что выполняется условие стабильности относительно главных операций:

$$\forall F \in \Omega_F \forall q_1, q_2, q_3, q_4 \in Q \left(\left((q_1 \overset{\sigma}{\equiv} q_3) \,\&\, (q_2 \overset{\sigma}{\equiv} q_4) \right) \Leftrightarrow \left(F(q_1, q_3) \overset{\sigma}{\equiv} F(q_2, q_4) \right) \right).$$

На множестве P два отношения будем считать эквивалентными при совпадении кортежей их именованных типов данных физического представления. Отношение эквивалентности σ на множестве Q\P введем в пункте 5, здесь же постулируем факт его существования. Фактор-множество Q/σ обозначим за W, а каноническое отображение Q→Q/σ за ω. Эквивалентность σ является также и конгруэнцией для алгебры **Q**, вследствие выполнения условия стабильности относительно главных операций, порождая фактор-систему **W=Q**/σ=<W,Ω_F>.

Требуется для каждого типа отношения $w \in W$ найти вектор – функцию D(w) размерности r, содержащую r1 независимых (экзогенных) переменных и r2 зависимых (эндогенных) (от $D_{r1}(w)$) переменных затрат ресурсов на получение отношения, r1 + r2 = r. Для w∈P/σ D(w) будет задана явно, исходя из характеристик данного отношения, и с учетом специфики его хранения на носителе. Независимыми переменными r1 отношения $w \in Q \setminus P/\sigma$, w=F($w_A$,$w_B$), $w, w_A, w_B \in W$ будем называть зависимые переменные $D_{r2}(w_A)$ и $D_{r2}(w_B)$, а также вектор собственных характеристик операции, известный для некоторых отношений типа w: $D_{r1}(q), q = F(q_A, q_B), q \in w, q_A \in w_A, q_B \in w_B$. $D_{r1}(q)$ может быть измерено, например, при выполнении запроса пользователя.

Будем искать D(w) для $w \in Q \setminus P/\sigma$ как

$$D_{r2}(w) = \Psi_w(D(w_A), D(w_B), D_{r1}(q)), \qquad (2)$$

т.е. определять каждый элемент r2 вектора D(w) как функцию от значений количественных характеристик операндов операции F, результатом которой является отношение q типа w, и статистической информации об операции F. В общем случае задача вычисления точного значения функции Ψ_q эквивалентна задаче выполнения запроса пользователя с точностью до последней операции $F_n \in \Omega_F$. Это очевидно следует из того факта, что до выполнения запроса точные значения $D(w_A)$ и $D(w_B)$ неизвестны.

Исходя из невозможности точного вычисления Ψ_q, оценим ее приближенно. Обозначим затраты некоторого заданного ресурса из (1) при обработке одного кортежа u как A. С точки зрения теории вероятностей A является событием. Отнесем наступлению события A единицу, ненаступлению – ноль. Предположим, что события повторны, независимы, и являются испытаниями Бернулли. Элементарным событием для n испытаний будет последовательность n нулей и единиц. Пространство элементарных событий для n испытаний Бернулли содержит 2^n точек. Вероятностной мерой события A является вероятность наступления p (вероятность ненаступления соответственно p_0=1-p). Вероятность того, что событие A наступит в испытаниях с определенными m номерами, а в остальных не наступит, равна $p^m q^{n-m}$. Вероятность того, что в n испытаниях событие A произойдет m раз $P_n(m) = C_n^m p^n q^{n-m} = b(m; n, p)$. Затратами ресурсов на обработку кортежей отношения q называется дискретная случайная величина ξ_q с биномиальным законом распределения $f_q(m)=b(m;n,p)$. Параметрами закона распределения являются количество событий n и вероятность p. Будем считать, что отношения одного типа w имеют одинаковый закон распределения $f_w(m)$. В общем случае закон распределения вероятности изменяется со временем, то есть $f_w(m)$ является функцией времени $f_w(m,t)$, однако, на данном этапе мы будем рассматривать систему как статическую на некотором интервале времени (t_1 , t_2), положив:

$$f_w(m) = \frac{\int_{t1}^{t2} f_w(m,t)dt}{t_2 - t_1}. \qquad (3)$$

Рассмотрим бинарную операцию, заданную на множестве пар кортежей (u_A, u_B) отношения $w_A \times w_B$. Затраты ресурсов на обработку кортежей отношения w_A характеризуются случайной величиной ξ_A, отношения w_B – ξ_B. Если рассматривать физическую реализацию алгоритмов в рамках теории асимптотических оценок количества операций, то обработка кортежей производится последовательно. В силу предположения о независимости событий A затраты ресурсов можно рассматривать как сумму двух разных испытаний Бернулли, $\xi_w = \xi_A + \xi_B$.

Искомые компоненты вектора Ψ_w – случайные величины ξ_w затрат ресурсов, законы распределения которых будут отличаться параметрами $p_{.,r}$ – вероятностью использования ресурса r и $n_.$ – количеством обработанных кортежей.

$$f_{\Psi_w}(m)_r = f_{\xi_A}(m, n_A, p_{A,r}) + f_{\xi_B}(m, n_B, p_{B,r}) \quad (4)$$

Значения $n_.$ и $p_{.,r}$ неизвестны. Так что вычисление $f_{\Psi_w}(m)_r$ на практике сильно затруднено. Известны значения затрат ресурсов для некоторых $q \in w$ (вектор $D_{r1}(q)$) и статистические данные системного каталога, содержащие агрегированные статистические показатели (выборочные средние). На их основе возможна оценка только математического ожидания случайной величины Ψ_w.

Обозначим $p_w = M[w]$. По свойствам математического ожидания:

$$M[\Psi_w]_r = M[\xi_{A,r}] + M[\xi_{B,r}] = n_A \cdot M[w_A]_r + n_B \cdot M[w_B]_r \quad (5)$$

С практической точки зрения n является количеством чтений значений кортежей данных отношения. Определим для операции $F \in \Omega_F$ асимптотическую функцию оценки количества действий при выполнении операции как $\Theta \in \Omega_\Theta$. Θ будем искать в виде $\Theta(D(w_A), D(w_B), D_{r1}(q))$. Не приводя сопутствующих рассуждений, постулируем, что значение n для всех типов ресурсов можно одинаково вычислить как площадь линейного многогранника V количества итераций, равную значению функции асимптотической оценки $\Theta \in \Omega_\Theta$ количества действий реляционной операции F:

$$n = \Phi \cdot \Theta(D(w_A), D(w_B), D_{r1}(q)), \quad (6)$$

Φ – константа, $\Phi \in R$

При непредельных значениях параметров D(w) операндов и операции оценку асимптотически точной функцией $\Theta \in \Omega_\Theta$ можно рассматривать как случайную величину

$$n = \Phi \cdot \Theta(D(w_A), D(w_B), D_{r1}(q)) + \overset{0}{\xi}, \quad (7)$$

где $\overset{0}{\xi}$ - случайная величина, характеризующая ошибку аппроксимации.

Минимизируем математическое ожидание $M[\overset{0}{\xi}]$ ошибки аппроксимации асимптотической функцией оценки Θ при непредельных значениях количественных характеристик. Для этого определим матрицу коэффициентов аппроксимации для операции F над двумя классами w_A и w_B Φ: $W \times W \to R^r$ асимптотической функцией оценки Θ затрат единиц ресурсов при выполнении операции как наилучшую аппроксимацию для Ψ_w относительно критерия минимума квадрата ошибки:

$$\Psi_w = \min \sum_i (\Phi \cdot \Theta[D(w_A), D(w_B), D_{r1}(q)] - D_{r2}(q)_i)^2, \quad (8)$$

где $D_{r2}(q)$ – известное значение вектора оценок. Как и в случае $D_{r1}(q)$ оно может быть измерено, например, при выполнении запроса пользователя.

4. Свойства асимптотической математической модели реляционного запроса

Исследуем свойства асимптотической оценки $\Theta \in \Omega_\Theta$ количества действий при выполнении операции $F \in \Omega_F$.

Физическая реализация реляционной операции $F \in \Omega_F$ в РСУБД есть некоторая циклическая конструкция [2] по обработке кортежей отношения с целью формирования нового отношения. Предположим, что все программы физической реализации принадлежат линейному классу [3] или же могут быть преобразованы к линейной программе существенно не уменьшая общности задачи. Тогда любая физическая реализация может быть описана минимальным графом зависимостей, заданном на пространстве итераций переменных цикла $V \in W \times W$. Функцию затрат ресурсов на выполнение операции будем рассматривать заданной на $W \times W$.

Поясним требование заданности функции на $W \times W$. Рассмотрим некоторую реляционную операцию $F \in \Omega_F$. Как показано в [4] любая операция множества Ω_F может быть представлена частным случаем элементарных операций алгебры A. Каждая операция в алгебре A трактуется как логическое реляционное отношение a. Таким образом, можно утверждать, что выполнение операции F эквивалентно проверке истинности логического реляционного высказывания. Проверка выполняется в общем случае для каждой пары кортежей $(u_A, u_B), u_A \in w_A, u_B \in w_B$. Заданная реляционная операция F как частный случай логического реляционного высказывания возможно будет рассматривать u_A и u_B на некоторых подмножествах $w`_A \subseteq w_A$ и $w`_B \subseteq w_B$ соответственно. При этом в силу линейности реализации [3] на множестве $W \times W$ рассматриваемые отношения будут заданы множеством линейных непересекающихся многогранников. Зададим в каждой точке (u_A, u_B) вероятность p использования ресурса r при

выполнении операции a. Затраты ресурсов как показано в (4) будут случайно величиной с биномиальным законом распределения, и $p(u_A,u_B) = p(u_A) + p(u_B)$. Тогда в силу того, что каждая точка просматривается не более одного раза, функция затрат ресурсов на выполнении операции F будет определяться соотношением (5), где параметр n определяется мощностью множества W×W как (6). Задание вероятности p в каждой точке (u_A,u_B) означает, что множество случайных событий составляет множество возможных пар (u_A,u_B). Рассматривая возможные типы ресурсов, это условие с практической точки зрения можно переформулировать следующим образом (необходимое условие): математическая модель строится для ресурсов, затраты которых зависят только от состава кортежей и не зависят, например, от порядка следования кортежей.

Множество V зависит от внешних переменных, определяющих размерности циклических конструкций и условия срабатывания операторов передачи управления. Внешние переменные известны и содержатся в строках векторов $D(w_A)$ и $D(w_B)$ и системном каталоге базы данных. В общем случае значения внешних переменных – случайные величины. Однако, на данном этапе мы будем рассматривать систему как статическую на некотором интервале времени (t_1 , t_2), положив по аналогии с (3)

$$D(q)_i = \frac{\int_{t1}^{t2} D(q,t)_i \, dt}{t_2 - t_1} \qquad (9)$$

Мощность неизвестного множества V будет асимптотически стремится к некоторой известной функции $\Theta[D(w_A), D(w_B), D_{r1}(q)]$. Для непредельных значений внешних переменных назовем функцию Θ оценочной, и будем находить ее неизвестные коэффициенты минимизируя квадрат ошибки (8).

Утверждение 1. *Математическое ожидание асимптотической аппроксимации унарной операции над отношением q обработки информации для реляционных СУБД имеет вид*

$$\Theta(\Psi_q) = M[w_A] * \Theta_q(D(w_A), D_{r2}(q)) \qquad (10)$$

Доказательство.

По теореме об информационном покрытии [3] минимальный граф зависимостей покрывается системой линейных функций, заданных на линейных многогранниках. Линейные многогранники описывают опорную область V пространства итераций отношения. Размеры и система линейных многогранников опорной области V зависит от внешних переменных $D(w_A)$ и $D_{r2}(q)$ и по построению в пределе:

$$|V| = \Theta_q(D(w_A), D_{r2}(q)) \qquad (11)$$

Использование ресурсов для чтения значения кортежа u на области V – элементарное событие, являющееся испытание Бернулли, и для унарной операции количество операций использования ресурсов будет характеризоваться случайной величиной ξ с биномиальным законом распределения

$$f(\xi) = f(m, |V|, p), \qquad (12)$$

где p - вероятность использования ресурса при выполнении операции с кортежем $u_A \in w_A$.

Обозначим p как $M[w_A]$, тогда характеризуя затраты ресурсов математическим ожиданием $\Theta(\Psi_q) = M[\xi] = M[w_A] * |V|$, или с учетом (11) $\Theta(\Psi_q) = M[w_A] * \Theta_q(D(w_A), D_{r2}(q))$.

Утверждение 2. *Математическое ожидание асимптотической аппроксимации бинарной операции над отношениями q_1 и q_2 обработки информации для реляционных СУБД имеет вид*

$$\Theta(\Psi_q) = M[w_A]\Theta_{q_1}(D(w_A), D(w_B), D_{r2}(q)) + M[w_B]\Theta_{q_2}(D(w_A), D(w_B), D_{r2}(q)) \qquad (13)$$

Доказательство утверждения аналогично доказательству утверждения 1. (12) для бинарных операций перепишется в виде:

$$f(\xi) = f_{\xi_A}(m, n_A, p_A) + f_{\xi_B}(m, n_B, p_B), \qquad (14)$$

где p_A - вероятность использования ресурса при выполнении операции с кортежем u_A;

p_B- вероятность использования ресурса при выполнении операции с кортежем u_B.

Обозначим p_A и p_B как $M[w_A]$ и $M[w_B]$ соответственно, количество операций по просмотру кортежей отношения w_A как $\Theta_{q_1}(D(w_A), D(w_B), D_{r1}(q))$, отношения w_B как $\Theta_{q2}(D(w_A), D(w_B), D_{r1}(q))$. Тогда характеризуя затраты ресурсов математическим ожиданием получим:

$$\Theta(\Psi_q) = M[w_A]\Theta_{q_1}(D(w_A), D(w_B), D_{r2}(q)) + M[w_B]\Theta_{q_2}(D(w_A), D(w_B), D_{r2}(q)). \qquad (15)$$

Что и требовалось доказать.

Исходя из свойств реляционной замкнутости, последовательность реляционных операторов можно представить в виде последовательности бинарных или унарных реляционных операций над результатами вложенных реляционных операций.

Следствие 1. *Математическое ожидание асимптотической аппроксимации бинарной реляционной операции F_3 получения производного отношения $q_C \in Q \setminus P$ из производных отношений q_A, $q_B \in Q \setminus P$ операций F_1 и F_2 соответственно имеет вид:*

$$\Theta(\Psi_{q_c}) = D(q_A)_M \Theta_{q_A}(\Psi_{q_A}, D_{r_2}(q_c)) + D(q_B)_M \Theta_{q_B}(\Psi_{q_B}, D_{r_2}(q_c)) \qquad (16)$$

Доказательство.

В формуле (15) для w_A, $w_B \in Q \setminus P$ применим (8), заменяя неизвестный вектор D(.) на известную функцию $\Psi(.)$. Математическое ожидание $M[w_A]$ производного отношения w_A – собственная характеристика отношения, показывающая трудоемкость получения одного элемента, - компонент номер M вектора $D(w_A)$. В результате эквивалентных замен получили (16).

Утверждение 3. *Возможно построение модели, в которой все операции над базовыми отношениями унарны.*

Доказательство.

Пусть в бинарную операцию входит базовое отношение. Построим новую реляционную операцию, произведя унарную операцию ограничения с заведомо истинным предикатом. Преобразование эквивалентно, однако уже не содержит бинарной операции над базовым отношением.

Следствие 2. *Возможно построение модели, в которой бинарные операции не содержат базовых отношений.*

Последовательно применяя операцию ограничения для всех базовых отношений, получим выражение, не содержащее бинарных операций над базовыми отношениями.

Утверждение 4. *Возможно построение модели, в которой все операции над производными отношениями бинарные.*

Доказательство. Пусть модель содержит унарную операцию F_j над производным отношением $q_{пр} \in Q \setminus P$. Введем новую операцию $F'_j = F_j(q_{iд}) \times 1$ как декартово произведение результата операции и отношения с нуль-арным заголовком, обозначенным как 1. Рассматривая 1 в качестве базового отношения, не содержащего кортежей, получим бинарную операцию, содержащую базовое отношение. Применив операцию ограничения над базовым отношением, получим утверждение 4.

Модели, содержащие только унарные операции над базовыми отношениями и только бинарные над производными отношениями, будем называть каноническими и в дальнейшем рассматривать только модели такого типа.

Выражение (16) задает рекуррентную формулу получения математического ожидания затрат ресурсов на получение отношения на основании известных количественных характеристик базовых отношений и известной асимптотически точной оценки операций $F_j \in \Omega_F$. Для удобства дальнейших рассуждений рекуррентное выражение будем раскрывать и обозначать как $\Psi_q` = M[\Psi_q] = \Psi_{q`}(q_i \mid q_i \in P)$.

Введем на семействе промежуточных отношений операцию лексикографического упорядочивания. Будем считать, что одно промежуточное отношение лексикографически меньше другого, если второе можно получить из первого путем каких-либо реляционных операций. Будем считать, что одна операция является родителем второй, если не существует операции, лексикографически больше первой и меньшей второй. Операции, получаемые посредством не более чем одной реляционной операции из отношения, будем называть потомками.

Лексикографически упорядоченная последовательность промежуточных отношений образует лексикографически упорядоченную последовательность $\Psi_q`$. Будем называть такую последовательность асимптотической аппроксимационной моделью запроса пользователя.

$$\begin{cases} & \\ & \\ & \\ & \\ & D(w_1) \\ & \ldots \\ & D(w_n) \\ D(w_{n+1}) = \Psi`_{w_{n+1}}(D(w_1),\ldots, D(w_n)) & \\ \ldots & \\ D(w_{n+m}) = \Psi`_{w_{n+m}}(D(w_1),\ldots, D(w_n)) & \end{cases} \qquad (17)$$

5. Эквивалентность производных отношений

Определим отношение эквивалентности на множестве производных отношений. Задача определения эквивалентности отношений представима в виде задачи преобразуемости конъюнктивного запроса [5].

Пусть задан конечный домен D, семейство отношений Q = (Q_1, Q_2, …, Q_m), где Q_i – подмножество d_i – мерных кортежей с элементами из D, множество выделенных переменных X, множество невыделенных переменных Y и два запроса $з_1$ и $з_2$ над X, Y, D и R. Запрос $з \in З = \{з_1, з_2\}$ имеет вид:

$$(x_1, x_2, \ldots, x_k)(\exists\, y_1, y_2, \ldots, y_l)(A_1 \wedge A_2 \wedge \ldots \wedge A_r)$$

для некоторых k, l и r, где $X` = (x_1, x_2, \ldots, x_k) \subseteq X$, $Y`= (y_1, y_2, \ldots, y_l) \subseteq Y$, и каждое A_i имеет вид $Q_j(u_1, u_2, \ldots, u_{dj})$, $u_s \in D \cup X` \cup Y`$, $1 \le s \le d_j$.

Тогда задачу преобразуемости конъюнктивного запроса можно сформулировать так: существует ли такая функция $\sigma : Y \to X \cup Y \cup D$, что если для каждого $y \in Y$ вместо любого вхождения y в $з_1$ подставить символ $\sigma(y)$, то в результате получится запрос $з_2$?

Данная задача полиномиально сводится к задаче изоморфизма графов. К данной задаче сводится задача раскрашивания графа в 3 цвета. Поэтому можно утверждать, что задача NP-трудна.

Требование строгой эквивалентности на практике в случае асимптотической модели и погрешности оценок можно ослабить. Будем считать, что 2 отношения эквивалентны, если они имеют одинаковое множество базовых отношений и одинаковое множество реляционных операций над ними.

Ослабление требования эквивалентности в некоторых случаях может привести к неадекватности модели. Если такое произошло, следует рассмотреть возможность усиления требования эквивалентности либо отказаться от объединения в (17) оценок эквивалентных отношений $q \in w$.

Список литературы

[1] Codd, E. F. A Relational Model of Data for Large Shared Data Banks // Communications of the ACM. – 1970. – Vol. 13, No. 6. – pp. 377-387.

[2] Конноли, Томас др. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд. – М.: Издательский дом “Вильямс”, 2000.

[3] Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.

[4] Date C. J., Darwen H. Foundation for Object/Relational Databases: The Third Manifesto (2d edition). – Reading, Mass.: Addison-Wesley, 2000.

[5] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи – М.: Мир, 1982.

Об одном обобщении взаимной корреляции и автокорреляции булевых функций

Погорелов Б. А., *профессор ИКСИ, г. Москва*
Пудовкина М. А., *доцент МИФИ, г. Москва*
E-mail: maricap@online.ru

Пусть $S(X)$ – симметрическая группа подстановок на множестве X, V_n – множество всех двоичных n-мерных векторов, e – тождественная подстановка, $<\alpha, x> = \sum_{i=1}^{n} \alpha_i x_i$, $\|f\|$ – вес функции f, $F_n = \{f : V_n \to \{0,1\}\}$, G_α – стабилизатор элемента α, $r_{f,g}(u) = \sum_{x \in V_n} (-1)^{f(x)+g(x+u)}$, $r_f(u) = r_{f,f}(u)$, где $u \in V_n$, $f, g \in F_n$, $l_\alpha(x) = <\alpha, x> = \sum_{i=1}^{n} \alpha_i x_i, \alpha, x \in V_n$, $w_f(\alpha) = \sum_{x \in V_n} (-1)^{<\alpha, x> + f(x)}$, o – нулевая функцию из F_n, т.е. $o(\alpha) = 0$ для всех $\alpha \in V_n$, H^T – транспонированная матрица.

Взаимная корреляция $r_{f,g}(u)$ и автокорреляция $r_f(u)$ булевых функций $f, g \in F_n$, $u \in V_n$, (см., например, [1]) являются важнейшим инструментом при исследовании криптографических свойств булевых функций. С их помощью можно описывать различные свойства булевых функций.

В работе рассмотрено обобщение взаимной корреляции и автокорреляции булевых функций относительно подгрупп группы $S(F_n)$. Также проведены обобщения равенства Парсеваля относительно групп AGL_n и $S_2 \uparrow S_n$.

Обобщение взаимной корреляции и автокорреляции относительно подгрупп группы $S(F_n)$

Для $H \le S(F_n)$, $x \in V_n$, $f, g \in F_n$, $h \in H$, положим

$\tilde{r}_{f,g}(h) = \sum_{x \in V_n} (-1)^{f(x)+g^h(x)}$, $\tilde{r}_{f,g}(H) = \sum_{h \in H} \tilde{r}_{fg}(h)$,

$\tilde{r}_f(h) = \tilde{r}_{f,f}(h)$, $\tilde{r}_f(H) = \tilde{r}_{f,f}(H)$,

$\tilde{\upsilon}_{H,f}(x) = \sum_{h \in H} (-1)^{f^h(x)}$.

Целочисленную функцию $\tilde{r}_{f,g}(h)$, определенную на группе $S(F_n)$, назовем взаимной h-корреляцией булевых функций $f, g \in F_n$, а целочисленную функцию $\tilde{r}_f(h)$ назовем h-автокорреляцией булевой функции $f \in F_n$.

Отметим, если группа H такая, что существует изоморфизм $\psi : H \to (V_n, +)$ и $g^h(x) = g(x + \psi(h))$ для любых $x \in V_n$, $g \in F_n$, то справедливо равенство $\tilde{r}_{f,g}(\psi^{-1}(u)) = r_{f,g}(u)$ для любого $u \in V_n$. Таким образом, взаимная h-корреляция булевых функций является естественным обобщением понятия взаимной корреляции функций, а h-автокорреляция – обобщение понятия автокорреляция функции.

Очевидно, что справедливо равенство $\tilde{r}_{f,g}(h) = \tilde{r}_{f,g^h}(e)$.

Пусть $H \le S_2 \uparrow S(V_n)$. Обозначим $\tilde{S}_n = \{(e,s) : s \in S(V_n)\}$, $\Omega_n = \{(\omega, e) \mid \omega \in F_n\}$, $\tilde{H} = H \cap \tilde{S}_n$, $\Omega_n(H) = \Omega_n \cap H$. Тогда $(\omega, s) : f(x) \to f(x^{s^{-1}}) + \omega(x^{s^{-1}})$.

Справедливы следующие свойства.

Утверждение 1. *Пусть $H \le S_2 \uparrow S(V_n)$, $\alpha \in V_n$. Тогда справедливы равенства*

$\tilde{\upsilon}_{H,f}(\alpha) = \mid \tilde{H}_\alpha \mid \sum_{\beta \in \alpha^{\tilde{H}}} \tilde{\upsilon}_{\Omega_n(H),f}(\beta)$, $\tilde{\upsilon}_{\Omega_n(H),f}(\alpha) = (-1)^{f(\alpha)} \tilde{\upsilon}_{\Omega_n(H),o}(\alpha)$, $\tilde{r}_{f,g}(H) = \sum_{\alpha \in V_n} (-1)^{f(\alpha)} \tilde{\upsilon}_{H,g}(\alpha)$.

Обозначим $r'_{f,g}(h) = \sum_{x \in V_n} (-1)^{f(x)+g(x^h)}$, $r'_{f,g}(H) = \sum_{h \in H} r'_{fg}(h)$, $r'_f(h) = r'_{f,f}(h)$, $r'_f(H) = r'_{f,f}(H)$,

$\upsilon'_{H,f}(x) = \sum_{h \in H} (-1)^{f(x^h)}$,

где $h \in H \le S(V_n)$, $x \in V_n$. Если $H \le S_2 \uparrow S(V_n)$, $\alpha \in V_n$ и $\Omega_n(H) = e$, то $\tilde{\upsilon}_{H,f}(\alpha) = \upsilon'_{\tilde{H},f}(\alpha)$, $\tilde{r}_{f,g}(H) = r'_{f,g}(\tilde{H})$ и $\tilde{r}_{f,g}(h) = r'_{f,g}(h^{-1})$.

Определим производную по подстановке $h \in S(V_n)$ как $D_h f(x) = f(x) + f(x^h)$.

Справедливы следующие свойства.

Утверждение 2. *Пусть $H \le S(V_n)$, $\alpha \in V_n$. Тогда справедливы равенства*

$\upsilon'_{H,f}(\alpha) = \mid H_\alpha \mid \sum_{\beta \in \alpha^H} (-1)^{f(\beta)}$,

$r'_{f,g}(H) = \sum_{\alpha \in V_n} (-1)^{f(\alpha)} \upsilon'_{H,g}(\alpha)$.

Следствие 3. *Пусть $H \le S(V_n)$ транзитивна, $\alpha \in V_n$. Тогда справедливы равенства*

$\upsilon'_{H,f}(\alpha) = \mid H_\alpha \mid \left(2^n - 2 \left\| f \right\|\right)$,

$$r'_{f,g}(H)=\left(2^n-2\|g\|\right)\sum_{\alpha\in V_n}(-1)^{f(\alpha)}\,|H_\alpha|\cdot$$

Если выполняется равенство $|H_\alpha|=|H_{\bar{0}}|$ *для всех* $\alpha\in V_n$, *то* $r'_{f,g}(H)=|H_{\bar{0}}|\left(2^n-2\|g\|\right)\left(2^n-2\|f\|\right)$.

Пусть группа $\hat{S}_n\le S_2\uparrow S(V_n)$ абстрактно изоморфна группе $S(V_n)$, где $\psi:\hat{S}\to S(V_n)$ соответствующий изоморфизм, такая, что $<\alpha,x>^{\psi^{-1}(h)}=<\alpha^h,x>$ для любого $h\in S(V_n)$. Тогда $\tilde{r}_{f,l_\alpha}(\psi^{-1}(h))=w_f(\alpha^h)$. При $h=e$ получаем коэффициент Уолша-Адамара.

Таким образом, коэффициенты Уолша-Адамара выражаются через взаимную e-корреляцией булевых функций $f,l_\alpha\in F_n$.

Для $H\le S(V_n)$ обозначим $\tilde{w}_H(\alpha,x)=\tilde{\upsilon}_{\psi^{-1}(H),l_\alpha}(x)$, т.е. $\tilde{w}_H(\alpha,x)=\sum_{h\in H}(-1)^{<\alpha^h,x>}$, $\alpha,x\in V_n$.

Утверждение 4. *Для произвольной подгруппы* $G\le S(V_n)$, *произвольных* $\alpha,x\in V_n$ *справедливо равенство*

$$\tilde{w}_G(\alpha,x)=|G_\alpha|\sum_{\beta\in\alpha^G}(-1)^{<\beta,x>},\ \sum_{h\in G}w_f(\alpha^h)=|G_\alpha|\sum_{\beta\in\alpha^G}w_f(\beta)\cdot$$

Следствие 5. *Для произвольной транзитивной подгруппы* $G\le S(V_n)$, *произвольных* $\alpha,x\in V_n$ *справедливы равенства*

$$\tilde{w}_G(\alpha,x)=\begin{cases}0,\ \text{при } x\neq 0;\\ 2^n|G_\alpha|,\ \text{при } x=0,\end{cases}$$

$$\sum_{h\in G}w_f(\alpha^h)=|G_\alpha|\sum_{\beta\in V_n}w_f(\beta)=2^n(-1)^{f(0)}|G_\alpha|\cdot$$

Обобщение равенства Парсеваля

Известно равенство Парсеваля (см., например, [1])

$$\sum_{u\in V_n}w_f(u)w_f(u+v)=\begin{cases}2^{2n},\ \text{при } v=0;\\ 0,\ \text{иначе.}\end{cases}$$

Приведем его обобщение относительно групп AGL_n и $S_2\uparrow S_n$.

Теорема 6. *Предположим, что* $x,y\in V_n$, $t,b\in AGL_n$, $h=t^{-1}b$ *и* $h=H+\varepsilon^T$, *где* $H=(h_{i,j})\in GL_n,\varepsilon\in V_n$. *Тогда справедливы равенства:*

1. $\sum_{\alpha\in V_n}(-1)^{<\alpha^t,x>+<\alpha^b,y>}=\begin{cases}2^n,\ \text{при } x=yH^T+<\varepsilon,y>;\\ 0,\ \text{иначе.}\end{cases}$
2. *(Обобщение равенства Парсеваля)*

$$\sum_{\alpha\in V_n}w_f(\alpha^t)w_f(\alpha^b)=2^n\sum_{y\in V_n}(-1)^{f(yH^T+<\varepsilon,y>)+f(y)}.$$

Утверждение 7. *Если* $h=(r,s)\in S_2\uparrow S_n$, $\tilde{s}=(e,s)$, *и* $\tilde{r}=(r_1,\dots,r_n)\in V_n$, *где* $\alpha^{r(i)}=\alpha+r_i$, *то выполняются равенства:*

1. $\sum_{\alpha\in V_n}(-1)^{<\alpha,x>+<\alpha^h,y>}=\begin{cases}2^n(-1)^{<\tilde{r}^{\tilde{s}},y>},\ \text{при } x_{s(i)}=y_i,\ i=1,\dots,n;\\ 0,\ \text{иначе.}\end{cases}$
2. *(Обобщение равенства Парсеваля):* $\sum_{\alpha\in V_n}w_f(\alpha)w_f(\alpha^h)=2^n\sum_{x\in V_n}(-1)^{f(x)+f(x^{\tilde{s}^{-1}})+<\tilde{r},x>}$.

Назовем булеву функцию $f:V_n\to\{0,1\}$ равновероятной относительно $h\in AGL_n$, если

$$\sum_{y\in V_n}(-1)^{f(yH^T+<\varepsilon,y>)+f(y)}=0.$$

Отметим, что булева функция f равновероятна относительно $h=(e,s)\in S_2\uparrow S_n$, если $\sum_{y\in V_n}(-1)^{f(y)+f(y^{s^{-1}})}=0$.

Следствие 8. *Пусть* $h\in AGL_n$. *Выполняется равенство* $\sum_{\alpha\in V_n} w_f(\alpha)w_f(\alpha^h)=0$ *тогда только тогда, когда функция* f *равновероятна относительно* h.

Утверждение 9

1. Если $h\in GL_n$, *то* $\sum_{\alpha\in V_n} w_f(\alpha)w_f(\alpha^h)=2^n r'_f(h^T)$.

2. Если $h=(e,s)\in S_2\uparrow S_n$, *то* $\sum_{\alpha\in V_n} w_f(\alpha)w_f(\alpha^h)=2^n r'_f(h^{-1})$.

Следствие 10 *Функция* $f\in F_n$ *равновероятна относительно*

1) $h\in GL_n$ *тогда только тогда, когда* $r_f'(h^T)=0$;

2) $h=(e,s)\in S_2\uparrow S_n$ *тогда только тогда, когда* $r_f'(h^{-1})=0$.

Список литературы.

1. О. А. Логачев, А. А. Сальников, В. В. Ященко. Булевы функции в теории кодирования и криптологии, МЦНМО, 2004.

О компьютерном моделировании логических процессов

Подколзин А. С., *доктор физико-математических наук, профессор кафедры МаТИС, механико-математический факультет МГУ им. Ломоносова*

Работа посвящена развитию технологии компьютерного моделирования логических процессов, возникающих при решении задач различного типа, в первую очередь математических. Процесс решения задачи разбивается на шаги, связанные с локальным планированием действий. Для каждого шага предлагается некоторая алгоритмическая процедура, способная выполнять его в аналогичных ситуациях и имеющая эвристический характер. Такие процедуры, называемые в работе приемами, накапливаются в компьютерном решателе задач. Они активизируются при решении новой задачи логическим процессором сканирующего типа, реализующим "внутреннее логическое зрение" системы.

Эффективность функционирования решателя обеспечивается проработкой большого числа обучающих примеров, на которых происходят необходимые коррекции логики принятия решений, аккумулируемой в приемах. Для записи приемов созданы два новых алгоритмических языка, радикально ускорившие и упростившие обучение системы. В настоящее время система насчитывает около 25000 приемов, позволяющих ей решать задачи по элементарным алгебре и геометрии, математическому анализу, аналитической геометрии, дифференциальным уравнениям, теории вероятностей. Система способна строить чертежи, сопровождающие вычислительные задачи по планиметрии. Входной информацией для построения чертежа служит формулировка задачи на логическом языке. Начато обучение системы в ряде других областей, в том числе таких, как элементарная физика и текстовые задачи, формулируемые на русском языке. Всего при обучении системы было проработано свыше 8000 примеров.

Решатель обеспечивает не только получение ответа задачи, но и подробный пошаговый показ ее решения. В этом отношении он выгодно отличается от существующих систем компьютерной математики и сам может рассматриваться как система компьютерной математики нового типа.

Следует подчеркнуть принципиальное различие целей, которые ставились при разработке решателя и целей, для которых были созданы коммерческие системы компьютерной математики. Решатель представляет собой инструмент для изучения логических процессов путем их моделирования. В этом качестве он необходим для получения ответа на главный вопрос - как научиться создавать саморазвивающиеся интеллектуальные системы. К сожалению, в этом отношении успехи сегодняшних технологий более чем скромны. Можно вспомнить времена, когда прогресс в искусственном интеллекте связывался с прогрессом в шахматных программах; предполагалось, что по мере усиления их игры

удастся обнаружить фундаментальные принципы разработки интеллектуальных систем. В наше время уже созданы исключительно мощные такие программы, способные обыгрывать чемпиона мира. Технология экспертных систем добилась впечатляющих результатов и во многих других областях. Во всех этих случаях мы имеем дело с чрезвычайно сложными узкоспециализированными алгоритмическими комплексами, зачастую разрабатывавшимися многими программистами, а иногда – поколениями программистов. Даже если предположить, что такого рода экспертные системы будут созданы практических для всех областей интеллектуальной деятельности человека и окажутся по своей эффективности превосходящими его возможности (что пока далеко не так), можно ли будет считать, что эта галерея экспертных систем и есть "настоящий" искусственный интеллект? Вероятно, нет, - ведь такая коллекция не ускорит научно-технический прогресс, а лишь зафиксирует его текущее состояние. Создается впечатление, что курс на разработку высокоэффективных экспертных систем для конкретных задач загоняет проблему изучения явления саморазвития алгоритмических комплексов в тупик. Программисты сразу начинают решать все проблемы организации такого комплекса "своим умом"; отвлекаться на самоанализ им, в общем-то, некогда, да и технологии здесь никакой не создано, и когда дело доходит до завершающих этапов, вся та сложнейшая интеллектуальная работа, которой и нужно было бы научить саму проектируемую систему, остается лишь в головах ее создателей. Участие в этом процессе компиляторов, выполняющих рутинные технические действия, ничуть не изменяет картины в целом.

Каким мог бы оказаться выход из данного тупика? Видимо, исходить нужно из того факта, что программы и алгоритмы имеют своим источником теоретические знания – будем их называть, для краткости, теоремами. Поэтому ключи к проблеме саморазвития интеллектуальных систем следует искать прежде всего в пограничной зоне между теоремами и алгоритмами, используя для программирования таких систем алгоритмический язык, максимально приближенный к языку теорем. Именно на этом пути и возник язык, применяемый в решателе. Прием решения задачи задается на нем как теорема предметной области, снабженная некоторой "алгоритмизирующей разметкой". Такая разметка определяет способ применения теоремы при решении задач и уточняет множество технических подробностей, позволяющих компилятору создать эффективную программу приема, фактически применяющего теорему. В описании приема выделяются два независимых логических уровня. Первый из них – уровень предметной области, т.е. формулировка теоремы. Второй – уровень структур данных, т.е. логическое описание условий на эти структуры, при которых применение приема, с точки зрения эксперта, является целесообразным. Разумеется, быстродействие систем, реализованных на языке такого высокого уровня и преобразующих информацию в громоздких, хотя и универсальных, логических структурах данных, будет существенно ниже, чем быстродействие систем, возникающих при обычном программировании. Этот недостаток преодолим, если продолжить цепочку программирования от той точки, где алгоритмы зарождаются, в направлении "нижнего" уровня. Систему можно научить компилировать свои медленные программы верхнего уровня в "обычные", по крайней мере для тех задач, где имеются простые структуры данных и быстрые алгоритмы. В этом она будет тоже имитировать действия человека – когда бывает нужно проделать какую-то объемистую вычислительную работу, он пишет программу для компьютера, но делает это с помощью своих логических процессов, пусть даже гораздо более медленных. Начатая в решателе работа по компиляции вычислительных процедур непосредственно с теоремного уровня, дополняющая компиляцию приемов, создает интересные перспективы обучения его смешанным логико-вычислительным процессам анализа математических моделей. С другой стороны, достоинством систем, обучаемых на языке высокого уровня, является дешевизна их программ и простота понимания смысла - ведь чтение таких программ, по существу, означает чтение теорем, из которых они извлечены.

Так как для подавляющего большинства предметных областей отсутствуют простые универсальные алгоритмы, то быстрое создание огромных массивов приемов, пусть даже и не очень эффективных в смысле быстродействия, но имеющих достаточно разумные решающие правила, создаст серьезные преимущества систем, программируемых "сверху" перед системами, программируемыми "снизу", просто за счет того, что они смогут охватить на порядки более обширные классы задач. Именно за счет такого "количественного" преимущества и удалось обучить решатель вычислительным задачам по планиметрии (пусть даже лишь на уровне стандартных задач средней сложности), в то время как распространенные системы компьютерной математики не имеют пока ничего похожего.

Разумеется, главным достоинством системы, обучаемой на языке, приближенном к языку теорем, является то, что она сама оказывается источником информации о принципах преобразования теорем в программы и принципах развития базы теорем, направленного на развитие алгоритмических возможностей. Это открывает перспективы для создания саморазвивающихся систем, способных решать нестандартные и творческие задачи.

В заключение автор хотел бы выразить свою искреннюю благодарность В. Б. Кудрявцеву за поддержку, сделавшую возможным проведение данного исследования.

Конечные автоматы в теории алгебраических моделей программ

Подловченко Р. И.,
Москва, НИВЦ МГУ
E-mail: rip@vvv.srcc.msu.su

В докладе рассматриваются алгебраические модели программ, при задании которых используются конечные автоматы. Дается обзор случаев, когда в таких моделях разрешима проблема эквивалентности.

Алгебраические модели программ введены в [1]. Опишем их.

Объектами модели являются схемы программ, построенные над двумя конечными алфавитами Y и P; элементы первого называются операторными символами, элементы второго — логическими переменными; последние принимают значения 0 и 1.

Структура схемы — это конечный ориентированный граф следующего вида. В нем выделены две вершины — вход без приходящих в него дуг и с одной исходящей и выход, не имеющий исходящих из него дуг. Остальные вершины имеют либо тип преобразователя, помеченного символом из Y и с одной исходящей из него дугой, либо тип распознавателя, помеченного символом из P и с двумя исходящими из него дугами, несущими метки 0 и 1.

Функционирование схемы осуществляется на функции разметки; каждая из них сопоставляет любой цепочке операторных символов из Y набор значений всех переменных из P. Множество таких функций обозначается $\mathcal{L}$. Выполнение схемы на функции μ из $\mathcal{L}$ представляет собой процесс обхода схемы, начинающийся в ее входе с пустой операторной цепочкой e и сопровождаемый ее накоплением. При этом переход через преобразователь с пометкой y сопровождается приписыванием символа y к текущей операторной цепочке справа, а переход через распознаватель с переменной p — выбором одной из исходящих из распознавателя дуг как продолжения трассы обхода; здесь берется дуга, помеченная значением p в наборе, который функция μ сопоставляет текущей операторной цепочке. Выполнение схемы завершается только с достижением ее выхода, и тогда накопленная операторная цепочка называется результатом выполнения.

Отдельная алгебраическая модель программ индуцируется двумя параметрами:

- эквивалентностью ν в множестве $Y^\star$ всех операторных цепочек;

- подмножеством L множества $\mathcal{L}$.

Параметры ν и L задают (ν, L)-эквивалентность схем. Она определяется так: две схемы (ν, L)-эквивалентны тогда и только тогда, когда, какой бы ни была функция разметки из L, всякий раз, как выполнение на ней одной из схем завершаемо, завершаемо и выполнение на ней другой схемы, и при этом результаты их выполнения — это цепочки, эквивалентные по отношению ν. Модель программ с (ν, L)-эквивалентностью принадлежащих ей схем называется (ν, L)-моделью.

В теории алгебраических моделей программ изучению подлежат только аппроксимирующие модели. Поясним суть отношения аппроксимации. Если каждый символ из Y заменить конкретным оператором, а каждый символ из P — конкретным предикатом, то схема превратится в соответствующую ей программу, а множество схем — в класс программ. В последнем традиционным образом определяется функциональная эквивалентность программ. Говорим, что модель аппроксимирует этот класс программ, если из эквивалентности схем в этой модели всегда следует функциональная эквивалентность соответствующих им программ.

В [2] получен достаточный признак того, что (ν, L)-модель является аппроксимирующей. Он заключается в требованиях:

1) отношение ν должно порождать полугруппу, элементами которой являются цепочки из $Y^\star$, а операцией — конкатенация цепочек, т.е. для любых h_1, h_2, h_3, h_4 из $Y^\star$ должно выполняться следующее:
$$h_1 \overset{\nu}{\sim} h_2 \;\&\; h_3 \overset{\nu}{\sim} h_4 \longrightarrow h_1 h_3 \overset{\nu}{\sim} h_2 h_4;$$
здесь записью $\overset{\nu}{\sim}$ обозначена эквивалентность цепочек по отношению ν;

2) множество L должно состоять из ν-согласованных функций разметки и быть замкнутым по операции сдвига на цепочку; при этом ν-согласованность функции разметки μ — это следующее ее свойство: для любых h_1, h_2 из $Y^\star$
$$h_1 \overset{\nu}{\sim} h_2 \longrightarrow \mu(h_1) = \mu(h_2),$$

а операция сдвига на цепочку h, применяемая к функции μ, дает, по определению, функцию μ', обладающую свойством:

$$\mu'(g) = \mu(hg),\ g \in Y^\star.$$

Множество ν-согласованных функций разметки обозначается $\mathcal{L}_\nu$.

В проблематике теории алгебраических моделей центральная роль принадлежит проблеме эквивалентности. Она состоит в поиске алгоритма, которые, получив на свой вход две произвольные схемы из выбранной модели, определяет, эквивалентны они или нет. При существовании такого алгоритма говорим, что в выбранной модели разрешима проблема эквивалентности.

Нами рассматриваются (ν, L)-модели, называемые автоматно-полугрупповыми. В них ν порождает полугруппу, а L представляет собой пересечение множества $\mathcal{L}_\nu$ с множеством, задаваемым автоматом. Такие автоматы введены в [3] под названием y_0-автоматов.

Каждый y_0-автомат строится над алфавитом $(Y \cup \{y_0\}) \times X$, где y_0 — добавочный к алфавиту Y символ, а X — множество всех наборов значений переменных из P. Язык, принимаемый y_0-автоматом, состоит из слов, называемых конфигурациями. Каждая конфигурация имеет вид

$$(y_0, x_0)(y_1, x_1)\dots(y_k, x_k),\ k \geq 0, \tag{1}$$

где для $i \geq 1$ $y_i \in Y$ и для $i \geq 0$ $x_i \in X$.

Любой функции разметки μ из L сопоставляется множество æ(μ) ассоциируемых с ней конфигураций. Полагаем, что конфигурация вида (1) принадлежит æ(μ) в том и только том случае, если

$$\mu(e) = x_0,\ \mu(y_1) = x_1,\ \dots,\ \mu(y_1 y_2 \dots y_k) = x_k.$$

Пусть A — y_0-автомат и $Q(A)$ — принимаемый им язык. Тогда, по определению, автомат A задает множество функций разметки $\mathcal{L}(A)$, определяемое следующим образом:

$$\mathcal{L}(A) = \{\mu | \mu \in \mathcal{L} \text{ и æ}(\mu) \subseteq Q(A)\}.$$

В [3] доказано, что имеет место лемма 1.

Лемма 1 *Существует алгоритм, который для любого y_0-автомата A определяет, является ли задаваемое им множество $\mathcal{L}(A)$ замкнутым по операции сдвига на цепочку.*

Этим в множестве автоматно-полугрупповых моделей программ выделены аппроксимирующие, ибо для любого ν, порождающего полугруппу, множество $\mathcal{L}_\nu$ замкнуто по операции сдвига на цепочку.

Опишем автоматно-полугрупповые (ν, L)-модели, для которых установлена разрешимость проблемы эквивалентности. При этом, поскольку $L = \mathcal{L}_\nu \cap \mathcal{L}(A)$, будем излагать требование к ν и $\mathcal{L}(A)$ и считать доказанными как существование y_0-автомата A, задающего рассматриваемое множество $\mathcal{L}(A)$, так и замкнутость $\mathcal{L}(A)$ по операции сдвига на цепочку.

Эти модели — следующие.

1. ν — тождество в $Y^\star$, $\mathcal{L}(A)$ — множество, задаваемое так называемым одношаговым автоматом A; оно замкнуто по операции сдвига на цепочку. Результат по разрешимости проблемы эквивалентности получен автором данного доклада и не опубликован.

2. ν — порождает полугруппу с левым сокращением и неразложимой единицей, $\mathcal{L}(A) = \mathcal{L}$. Этот случай рассмотрен в [4].

3. ν порождает полугруппу с константами, $\mathcal{L}(A) = \mathcal{L}$. Случай, когда константа одна, рассмотрен в [5], общий случай — в [6].

4. ν — свободно коммутативная эквивалентность, $\mathcal{L}(A)$ состоит из функций разметки, индуцируемых сопоставлением каждому символу y из Y подмножества sy множества P, причем:

$$y_1 \neq y_2 \ \longrightarrow\ sy_1 \cap sy_2 = \emptyset,\ y_1, y_2 \in Y.$$

По определению, $\mu \in \mathcal{L}(A)$ тогда и только тогда, когда для любых h из $Y^\star$, y из Y и p из P выполнено требование:

$$\mu(h)|_p = \mu(hy)|_p,\ \text{если } p \in P \setminus sy.$$

Здесь и далее $\mu(h)|_p$ — это значение переменной p в наборе $\mu(h)$.

В [7] установлено, что проблема эквивалентности в данной модели сводится к проблеме эквивалентности многоленточных автоматов; разрешимость последней установлена в [8].

5. ν — частично коммутативная эквивалентность; $\mathcal{L}(A)$ состоит из функций разметки, индуцируемых выбором подмножеств P_1, P_2 множества P и удовлетворяющих требованиям: для любых h из $Y^\star$ и y из Y

$$\mu(h)|_p = 0 \longrightarrow \mu(hy)|_p = 0, \text{ если } p \in P_1;$$
$$\mu(h)|_p = 1 \longrightarrow \mu(hy)|_p = 1, \text{ если } p \in P_2.$$

Этот случай рассмотрен в [9].

6. ν — частично-коммутативная эквивалентность, обладающая свойством транзитивности $\mathcal{L}(A)$ состоит из монотонных функций разметки. По определению, функция μ из $\mathcal{L}$ называется монотонной, если при любых h из $Y^\star$, y из Y и p из P

$$\mu(h)|_p \leq \mu(hy)|_p.$$

Для данного случая результат по разрешимости проблемы эквивалентности не опубликован. Он получен использованием алгоритма, описанного в [10] для случая свободно коммутативной эквивалентности.

Список литературы

1. Подловченко Р.И. Иерархия моделей программ // Программирование, 1981, є 2, с. 3-14.

2. Подловченко Р.И. Полугрупповые модели программ // Программирование, 1981, є 4, с. 3-13.

3. Подловченко Р.И., Аланакян Н.А. Регулярные модели программ // Программирование, 1993, є 4, с. 3-11.

4. Глушков В.М., Летичевский А.А. Теория дискретных преобразователей // Избранные вопросы алгебры и логики, Новосибирск, Наука, 1973, с. 5-39.

5. Подловченко Р.И., Русаков Д.М. Схемы программ с константами // Программрование, 2005, є 3, с. 5-18.

6. R. Podlovchenko, D. Rusakov, V. Zakharov. On the equivalence problem for programs with mode switching // Lecture Notes in Computer Science, v. 3845, 2006, p. 351-352.

7. Подловченко Р.И., Алгебраические модели программ и автоматы // Математические вопросы кибернетики, вып. 15, 2006.

8. Harju T., Karhumaki J. The equivalence of multi-tape finite automata // Theoretical Computer Science, 1991, 78, p. 347-355.

9. Захаров В.А., Захарьящев И.М. Об эквивалентности программ с частично перестановочными операторами, сохраняющими значения предикатов // Труды VI Международной конференции “Дискретные модели в теории управляющих систем”, Москва, 7-11 декабря 2004, изд. отдел фак. ВМиК МГУ, 2004.

10. Подловченко Р.И. О схемах программ с перестановочными и монотонными операторами // Программирование, є 5, 2003, с. 46-54.

Критерий нелинейной однослойности нейронных схем

Половников В. С.,

кафедра математической теории интеллектуальных систем
механико-математического факультета МГУ им. М. В. Ломоносова

В работе [1] доказано, что для любой нейронной схемы без памяти существует эквивалентная ей схема нелинейной глубины не более 2. Также показано, что существуют кусочно-линейные функции, для реализации которых нейронными схемами единичной нелинейной глубины недостаточно. В данной работе исследуются функции, представимые нейронными схемами глубины 1 и приводится критерий представимости кусочно-линейных функций такими схемами.

Дискретные нейронные сети рассматриваются с точки зрения функциональных систем [2]. За счет расширения функционального базиса было предложено более общее, чем нейронная сеть, понятие нейронной схемы [1, 3].

Пусть $f(\bar{x})$ — кусочно-линейная функция $\mathbb{R}^n \to \mathbb{R}$, заданная гиперплоскостями $l_i = \{\bar{x} \in \mathbb{R}^n | \bar{a}_i \cdot \bar{x} + c_i = 0\}$, $i = 1, \ldots, k$, которые разбивают $\mathbb{R}^n$ на классы эквивалентности $R_1, \ldots, R_s$, т.е. на подмножества $\mathbb{R}^n$, для которых векторы сигнатуры [1, 3] совпадают. Согласно определению данному

в [1] и [3], существуют линейные функции $f_{R_1}, \ldots, f_{R_s}$, такие, что $f_{R_j} : \mathbb{R}^n \to \mathbb{R}$, $f_{R_j}(\bar{x}) = \bar{b}_j \cdot \bar{x} + d_j$, $j = 1, \ldots, s$, и для $\bar{x}$ из R_j выполнено соотношение $f(\bar{x}) = f_{R_j}(\bar{x})$.

Тройку классов эквивалентности (R_+, R_0, R_-), $R_+, R_0, R_- \in \{R_1, \ldots, R_s\}$, таких, что $\sigma_+^t = \sigma_0^t = \sigma_-^t$ при $t = 1, \ldots, i-1, i+1, \ldots, n$, а $\sigma_+^i = 1$, $\sigma_0^i = 0$ и $\sigma_-^i = -1$, назовем *смежной тройкой* к гиперплоскости l_i, где $\sigma(R_+) = (\sigma_+^1, \ldots, \sigma_+^n)$, $\sigma(R_0) = (\sigma_0^1, \ldots, \sigma_0^n)$ и $\sigma(R_-) = (\sigma_-^1, \ldots, \sigma_-^n)$ — векторы сигнатур классов R_+, R_0 и R_-. А сами классы R_+, R_0, R_- назовем *соседними* к гиперплоскости l_i.

Если гиперплоскость l не совпадает ни с одной из гиперплоскостей $l_1, \ldots, l_k$, задающих f, то будем считать, что тройка $(l, 0, 0)$ является *переходом* для f через l. Если для гиперплоскости l_i функции f найдутся линейные функции $l_{+;i}(\bar{x}) = \bar{a}_{+;i}\bar{x} + c_{+;i}$ и $l_{-;i}(\bar{x}) = \bar{a}_{-;i}\bar{x} + c_{-;i}$, такие, что для всех троек (R_+, R_0, R_-), смежных с l_i, выполняются равенства $f_{R_+} - f_{R_0} = l_{+;i}$ и $f_{R_-} - f_{R_0} = l_{-;i}$, то тройка $(l_i, l_{+;i}, l_{-;i})$ называется переходом f через гиперплоскость l_i. В противном случае будем говорить, что переход через гиперплоскость l_i не существует. Переход называется *нетривиальным*, если хотя бы одна из двух его последних компонент отлична от нуля, иначе — *тривиальным*.

Таким образом, у кусочно-линейной функции, которую можно задать при помощи k гиперплоскостей, существует не более k различных нетривиальных переходов.

Теорема 1 *Пусть кусочно-линейная функция f задана гиперплоскостями $l_1, \ldots, l_k$. Если переход через l_i тривиален, то f можно задать без использования гиперплоскости l_i.*

Следствие 1 *Функция, для которой переходы через любую гиперплоскость тривиальны, линейна.*

Предложение 1 *Пусть кусочно-линейные функции f^1 и f^2 заданы гиперплоскостями $l_1, \ldots, l_k$ и для этих функций существуют переходы через каждую гиперплоскость l_i, $i = 1, \ldots, k$ (необязательно нетривиальные). Пусть $\{(l_1, l^1_{+;1}, l^1_{-;1}), \ldots, (l_k, l^1_{+;k}, l^1_{-;k})\}$ — множество переходов функции f_1, а $\{(l_1, l^2_{+;1}, l^2_{-;1}), \ldots, (l_k, l^2_{+;k}, l^2_{-;k})\}$ — множество переходов функции f_2. Тогда функция $f = f^1 + f^2$ является кусочно-линейной и задана гиперплоскостями $l_1, \ldots, l_k$. Для нее также существуют переходы через любую гиперплоскость, причем набор переходов через $l_1, \ldots, l_k$ для f следующий: $\{(l_1, l^1_{+;1} + l^2_{+;1}, l^1_{-;1} + l^2_{-;1}), \ldots, (l_k, l^1_{+;k} + l^2_{+;k}, l^1_{-;k} + l^2_{-;k})\}$.*

Нетрудно видеть, что для любых кусочно-линейных функций f^1 и f^2 множество задающих их гиперплоскостей можно считать одним и тем же, если объединить множества гиперплоскостей, задающих f^1 и f^2.

Лемма 2 *Пусть $\bar{x} \in \mathbb{R}^n$. Если кусочно-линейные функции $f^1(\bar{x})$ и $f^2(\bar{x})$ имеют одинаковые множества нетривиальных переходов, то они различаются только на линейную функцию. Т.е. для некоторых $\bar{c} \in \mathbb{R}^n$ и $d \in \mathbb{R}$ выполнено равенство $f^1(\bar{x}) = f^2(\bar{x}) + \bar{c} \cdot \bar{x} + d$.*

Нелинейной глубиной нейронной схемы называется наибольшее количество нелинейных элементов (θ или F) на путях, ведущих от какого-либо входа схемы к ее выходу. Подробно определение нелинейной глубины дано в [1, 3].

Теорема 2 *Кусочно-линейная функция представима нейронной схемой нелинейной глубины 1 с некоторой нелинейной сложностью p тогда и только тогда, когда у нее существуют переходы через любую гиперплоскость, причем всего в этих переходах не более p ненулевых вторых или третьих компонент.*

Следствие 2 *Кусочно-линейную функцию f можно реализовать нейронной схемой нелинейной глубины 1 тогда и только тогда, когда существует такое натуральное число k, что f представима в виде*

$$f(\bar{x}) \;=\; \sum_{i=1}^{k} (F(\bar{a}_{+;i}\bar{x} + c_{+;i}, \bar{a}_i \cdot \bar{x} + c_i) + F(\bar{a}_{-;i}\bar{x} + c_{-;i}, -\bar{a}_i \cdot \bar{x} - c_i)) + \bar{b} \cdot \bar{x} + d,$$

для некоторых $\bar{a}_{+;i}, \bar{a}_{-;i}, \bar{b} \in \mathbb{R}^n$ и $c_{+;i}, c_{-;i}, d \in \mathbb{R}$, $i = 1, \ldots, k$.

Список литературы

1. Половников В. С. О некоторых характеристиках нейронных схем. // Интеллектуальные системы. 2004 8, вып. 1-4. 121–145.

2. Кудрявцев В. Б. Функциональные системы. М.: Изд-во МГУ, 1982.

3. Половников В. С. О некоторых характеристиках нейронных схем. // Вестн. Моск. ун-та. Матем. Механ. 2004. №5. 65–67.

Оптимизация универсального тестирования поведения автоматов

Пономаренко А. В.,
Институт проблем точной механики и управления РАН
410208, г. Саратов, ул. Рабочая, д. 24, ИПТМУ РАН
тел.: (8452) 764169; e-mail: disciple-alex@nemail.ru

В работе Твердохлебова В. А. [1] разработано понятие универсального тестирования конечных детерминированных автоматов. В дальнейших работах Твердохлебова В. А. продолжено исследование универсальных тестов - найдены оценки длины и методы построения универсальных тестов. Итоговые результаты работ по универсальному тестированию Твердохлебов В. А. приведены в работе [2]. Универсальные тесты не зависят от свойств функций переходов и выходов автоматов распознаваемого семейства (в отличие от частных тестов, построенных по методу А. Гилла [3]), а зависят только от мощности входного алфавита |X| и от n – максимального числа состояний для любого автомата семейства, являющегося параметром теста. Полученная Твердохлебовым В. А. оценка длины универсального теста $p \in X^*$:

$$|p| \le |X|^{n^2+n-1} + n^2 + n - 2, \quad (1)$$

показала неэффективность их практического применения. В докладе рассматривается вопрос о возможности сокращения длин универсальных тестов в классах конечных детерминированных автоматов, комбинационные части которых обладают специфическими свойствами.

Для выделения частных классов в классе конечных детерминированных автоматов используется разработанная в работе [4] классификация автоматов на основе фундаментальных свойств Поста. В основе классификации автоматов, лежит известная декомпозиция автомата на комбинационную часть, рассматриваемую как совокупность функций алгебры логики, и память. Для функций алгебры логики имеется фундаментальная классификация по свойствам Поста – сохранять нуль, сохранять единицу, быть линейной, быть самодвойственной, быть монотонной. Для любой булевой функции и любого свойства Поста существуют эффективные процедуры проверки, обладает ли функция рассматриваемым свойством. В связи с этим всего возможно 32 варианта сочетаний свойств Поста. Для обозначения этих классов будем использовать букву H с пятью нижними индексами: $H_{abcde} = H_a \cap H_b \cap H_c \cap H_d \cap H_e$,где $a = 0 \mid \overline{0}, \quad b = 1 \mid \overline{1}, \quad c = S \mid \overline{S}, \quad d = L \mid \overline{L}, \quad e = M \mid \overline{M}$. Также рассматриваются более широкие классы, например, H_0, H_{0L} и т.д. Для классификации автомата его функции переходов и выходов δ и λ представляются в виде наборов булевых функций $\delta = <\delta_1, \delta_2, ..., \delta_\omega>, \ \lambda = <\lambda_1, \lambda_2, ..., \lambda_\nu>$. Отнесение автомата к какому-либо классу автоматов происходит на основе принадлежности $\delta_i, 1 \le i \le \omega$ и $\lambda_j, 1 \le j \le \nu$ какому-либо классу функций. Так, например, будем считать, что автомат является линейным $A \in H_L$, если все булевы функции из наборов, составляющих функцию переходов и функцию выходов – линейные, $\lambda_j \in H_L$ и $\delta_i \in H_L$.

Для исследования возможности понижения оценки длины универсального теста был выбран класс (4,2,2)-автоматов вида $A = ((E_2)^2, E_2, E_2, \delta, \lambda)$, где $E_2 = \{0,1\}$. Этот класс конечных детерминированных автоматов содержит 16777216 автоматов, включая эквивалентные автоматы. В исследуемом классе автоматов для проведения вычислительного эксперимента были выделены подклассы автоматов: H_L , H_S , H_M , H_{SL} , H_{SM} , H_{LM} , H_{SLM} . Для выделенных подклассов класса конечных детерминированных (4, 2, 2) – автоматов вычислено общее число автоматов, число попарно неэквивалентных автоматов и определены пары автоматов, составляющие исключительные классы (см. таблицу 1). Определение пар составляющих исключительные классы проводилось по известному алгоритму проверки эквивалентности состояний (см. А. Гилл [3], стр. 94-95)

Таблица 1. Характеристики исследуемых классов

Класс	Кол-во авт.	Кол-во попарно неэквивалентных авт.	Кол-во пар, составляющих исключительный класс
H_L	4096	84	3378
H_S	4096	356	61384

H_M	8000	1460	621212
H_{SL}	512	27	341
H_{SM}	64	13	42
H_{LM}	125	10	30
H_{SLM}	27	3	2

Универсальный тест в классе (4,2,2)- автоматов вида $A=((E_2)^2, E_2, E_2, \delta, \lambda)$, где $E_2 = \{0,1\}$, по оценке (1) имеет длину 524 306 символов. Для проверки возможности уменьшения длины универсального теста были выбраны два способа сокращения: суффиксное сокращение – определяется префикс первоначального теста, сохраняющий свойство универсальности в исследуемом подклассе; префиксное сокращение – определяется суффикс первоначального теста, сохраняющий свойство универсальности в исследуемом подклассе. Для проведения вычислительного эксперимента были построены два варианта универсального теста:

Вариант 1. На множестве $X = \{0,1\}$ порядок "$\prec$" задается по правилу – $0 \prec 1$.

Вариант 2. На множестве $X = \{0,1\}$ порядок "$\prec$" задается по правилу – $1 \prec 0$.

В результате проведения вычислительного эксперимента в выбранных подклассах класса (4,2,2)-автоматов было установлено значительное сокращение оценки длины универсального теста.

Утверждение 1. *В подклассе линейных автоматов H_L класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 24 символа; префикс 2-го варианта универсального теста длиной 24 символа; суффикс 1-го варианта универсального теста, длиной 22 символа; суффикс 2-го варианта универсального теста длиной 23 символа.*

Утверждение 2. *В подклассе самодвойственных автоматов H_S класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 117 символа; префикс 2-го варианта универсального теста, длиной 117 символа; суффикс 1-го варианта универсального теста, длиной 106761 символа; суффикс 2-го варианта универсального теста длиной 106761 символа.*

Утверждение 3. *В подклассе монотонных автоматов H_M класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 533 символа; префикс 2-го варианта универсального теста длиной 533 символа; суффикс 1-го варианта универсального теста, длиной 106765 символа; суффикс 2-го варианта универсального теста длиной 106765 символа.*

Утверждение 4. *В подклассе самодвойственных линейных автоматов H_{SL} класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 23 символа; префикс 2-го варианта универсального теста длиной 23 символа; суффикс 1-го варианта универсального теста, длиной 21 символа; суффикс 2-го варианта универсального теста длиной 22 символа.*

Утверждение 5. *В подклассе самодвойственных монотонных автоматов H_{SM} класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 58 символа; префикс 2-го варианта универсального теста длиной 58 символа; суффикс 1-го варианта универсального теста, длиной 9368 символа; суффикс 2-го варианта универсального теста длиной 9368 символа.*

Утверждение 6. *В подклассе линейных монотонных автоматов H_{LM} класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта универсального теста, длиной 22 символа; префикс 2-го варианта универсального теста длиной 22 символа; суффикс 1-го варианта универсального теста, длиной 20 символа; суффикс 2-го варианта универсального теста длиной 21 символа.*

Утверждение 7. *В подклассе самодвойственных линейных и монотонных автоматов H_{SLM} класса (4,2,2)-автоматов сохраняет свойство быть универсальным тестом: префикс 1-го варианта*

универсального теста, длиной 22 символа; префикс 2-го варианта универсального теста длиной 22 символа; суффикс 1-го варианта универсального теста, длиной 19 символа; суффикс 2-го варианта универсального теста длиной 19 символа.

Далее в докладе рассматривается вопрос о возможности использования минимизированных тестов для распознавания автомата из исследуемых подклассов, являющегося элементом композиции автоматов из исследуемых подклассов, а также для распознавания всей композиции в целом. Для этого исследуется возможность трансляции универсальных тестов в классе (4, 2, 2)- автоматов и минимизированных универсальных тестов в классе линейных (4, 2, 2)-автоматов компонентами структурных автоматов, в случае если они принадлежат специфическим классам конечных детерминированных автоматов.

Структурная теория автоматов приведена в работе [5] В. М. Глушкова. В структурной теории автоматов, в отличие от абстрактной, учитывается внутренняя структура автомата и структура его входных и выходных сигналов. Под автоматом в структурной теории понимается абстрактный автомат с явно выделенными и пронумерованными элементарными входными и выходными каналами с соответствующими входными и выходными узлами. Переход от абстрактных входных и выходных сигналов к структурным, происходит путем кодирования, то есть сопоставлением абстрактному сигналу его записи в структурном алфавите автомата. Основным направлением исследования является композиция автоматов, то есть метод построения из относительно простых более сложных автоматов. В работе В. М. Глушкова описан общий способ и приведены правила композиции автоматов.

Определение 1. Будем говорить, что инициальный автомат (A, s_0), где $A = (S, X, Y, \delta, \lambda)$ транслирует универсальный тест p, если $q = \lambda(s_0, p)$ является универсальным тестом.

В результате проведения вычислительного эксперимента были получены следующие результаты.

Утверждение 8. *Из 336 попарно неэквивалентных инициальных линейных (4,2,2) – автоматов 10 транслируют 1-й вариант универсального теста*.

Утверждение 9. *Из 1424 попарно неэквивалентных инициальных самодвойственных (4,2,2) – автоматов 76 транслируют 1-й вариант универсального теста.*

Утверждение 10. *Из 5840 попарно неэквивалентных инициальных монотонных (4,2,2) – автоматов 81 транслируют 1-й вариант универсального теста.*

Утверждение 11. *Из 336 инициальных линейных (4, 2, 2) – автоматов 214 транслируют минимизированный универсальный тест для линейных (4, 2, 2) – автоматов.*

Утверждение 12. *Из 1424 инициальных самодвойственных (4, 2, 2) – автоматов 688 транслируют минимизированный универсальный тест для линейных (4, 2, 2) – автоматов.*

Утверждение 13. *Из 5840 инициальных монотонных (4, 2, 2) – автоматов 1479 транслируют минимизированный универсальный тест для линейных (4, 2, 2) – автоматов.*

Таким образом, в докладе показано значительное (для линейных автоматов – на 5 порядков, для самодвойственных автоматов - на 5 порядков, для монотонных автоматов - на 4 порядка) сокращение универсальных тестов при тестировании автоматов из специальных классов. Так же рассмотрена возможность трансляции универсальных тестов автоматами из специальных классов и показано, что ни одно из выбранных свойств автоматов по используемой классификации – линейность, самодвойственность или монотонность не являются достаточными для трансляции универсальных тестов. Кроме этого выделены (4, 2, 2) – автоматы, любая (по длине) последовательная композиция которых может тестироваться универсальным тестом для (4,2,2) – автоматов.

Список литературы

1. Твердохлебов В. А. Универсальные генераторы тестов и системы диагностирования // Техническая диагностика. – Ростов-на-Дону, 1982.

2. Твердохлебов В. А. Методы построения универсальных тестов для конечных автоматов. // «Автоматика и Телемеханика» №1, с. 154-163, Москва, 2005.

3. А. Гилл. Введение в теорию конечных автоматов. // Москва: Изд-во «Наука», 1966.

4. Твердохлебов В. А., Пономаренко А. В. Классификация конечных автоматов по свойствам функций переходов и выходов //Сборник научных трудов ИПТМУ РАН, Саратов, изд-во СГТУ, 2004. с. 16-25.

5. Глушков В. М.. Синтез цифровых автоматов // М.: Физматгиз, 1962. 162 с.

О проблеме расшифровки ДНК гибридизацией

Попов В. Ю.,
профессор кафедры алгебры и дискретной математики
620083, Уральский государственный университет им. А.М. Горького
E-mail: Vladimir.Popov@usu.ru

Лабораторное оборудование на сегодняшний день не позволяет анализировать последовательность ДНК полностью. Поэтому ДНК разделяют на фрагменты. После этого возникает проблема расшифровки, т.е. восстановления исходной последовательности по имеющимся фрагментам. Одним из основных методов расшифровки является расшифровка гибридизацией (см., например, работу [1] и библиографию в этой работе). При расшифровке гибридизацией гибридизируются все возможные олигонуклеотиды фиксированной длины k из данной последовательности, потом, используя попарные наложения, определяется множество X, обеспечивающее реконструкцию исходной последовательности. Например, последовательность $ATCCGC$ может быть реконструирована из множества

$$X = \{ATC, TCC, CCG, CGC\}.$$

Расшифровка гибридизацией обычно формализуется в терминах правильного наложения. Говорят, что строка s получена наложением длины $k1$ пар строк из X, если для любой подстроки t длины k строки s имеет место соотношение $t \in X$.

Дан алфавит $\Gamma = \{A, C, T, G\}$ и множество $X \subseteq \Gamma^k$, т.е. $X \subseteq \Gamma^*$ и $|x| = k$ для каждой строки $x \in X$, правильным наложением s множества X называется строка s такая, что каждый элемент $x \in X$ появляется по крайней мере один раз как подстрока строки s и s получена наложением длины $k-1$ пар строк из X. Полиномиальный алгоритм реконструкции при расшифровке гибридизацией известен только в случае, когда известно количество вхождений для каждого элемента $x \in X$ и существует единственное правильное наложение множества X [2].

В работе [3] предложен комплексный подход к проблемам вычислительной сложности, связанным с расшифровкой гибридизацией, и, в частности, рассмотрена следующая проблема:

Проблема кратчайшей реконструкции с дополнениями при расшифровке гибридизацией (SBH-ADD)

Дано: *Натуральное число k, конечное множество X строк из Γ^k, натуральные числа L, m.*

Вопрос: *Существует ли множество Y строк из Γ^k, $|Y| < m+1$, такое, что существует правильное наложение s множества $X \cup Y$, длина которого не превосходит L?*

В [3] поставлен вопрос о вычислительной сложности проблемы SBH-ADD при фиксированном значении параметра k:

Является ли проблема SBH-ADD *параметрически разрешимой относительно параметра k, т.е. существует ли алгоритм, решающий проблему* SBH-ADD *за время $f(k)g(n)$, где $f(k)$ — произвольная функция, зависящая только от k, а $g(n)$ — полином, зависящий от размера исходных данных n?*

Положительный ответ на этот вопрос дает следующая теорема.

Теорема. *Проблема* SBH-ADD *является параметрически разрешимой относительно параметра k.*

В работе [4] установлено, что проблема SBH-ADD является **NP**-полной. Поэтому не следует надеяться на нахождение быстрого алгоритма, решающего проблему SBH-ADD в общем случае. Поскольку обычно $k \leq 12$, один из практических подходов к решению проблемы SBH-ADD дает приведенная выше теорема, согласно которой, существует алгоритм, полиномиально зависящий от размера исходных данных и позволяющий для сравнительно небольших значений k находить оптимальное правильное наложение. Кроме того, в случае, когда значение параметра k достаточно велико для массового применения этого алгоритма, его можно использовать для создания обучающей выборки, при помощи которой может быть создана эффективная нейронная сеть, строящая правильное наложение. С учетом того, что, как правило, значение параметра L, хотя и достаточно велико, изменяется в ограниченном диапозоне, для решения проблемы SBH-ADD может применяться не только рекуррентная нейронная сеть, строящая правильное наложение частями (например, сеть Эльмана), но и многослойная однонаправленная.

В качестве другого естественного пути решения проблемы SBH-ADD можно рассматривать генетический алгоритм, интерпретирующий SBH-ADD как проблему ограниченной сложности (общую информацию по генетическим алгоритмам можно найти, например, в [5]).

Функция приспособленности $f(x_1, x_2, \ldots, x_s)$, где $\{x_1, x_2, \ldots, x_s\}$ — множество переменных, принимающих значения на генах хромосомы, называется аддитивно разложимой, если существует семейство функций $F = \{f_1(x_{i_{1,1}}, x_{i_{1,2}}, \ldots, x_{i_{1,p_1}}), \ldots f_t(x_{i_{t,1}}, x_{i_{t,2}}, \ldots, x_{i_{t,p_t}})$, где $p_j << s$ для любого

значения j, такое, что

$$f(x_1, x_2, \ldots, x_s) = \sum_{j=1}^{t} f_j(x_{i_{j,1}}, x_{i_{j,2}}, \ldots, x_{i_{j,p_t}}).$$

При этом функции семейства F называются частичными функциями приспособленности. Функция приспособленности называется полусепарабельной, если для любой переменной x_j количество частичных функций приспособленности, завилящих от этой переменной, много меньше общего количества частичных функций приспособленности.

Проблема, решаемая генетическим алгоритмом, называется проблемой ограниченной сложности, если функция приспособленности является аддитивно разложимой и полусепарабельной и частичные функции приспособленности имеют одинаково распределенные значения (см. [6]). Рассмотрение проблем ограниченной сложности представляет интерес, поскольку для этих проблем генетический алгоритм с высокой вероятностью выдает оптимальное решение за ограниченное время (см., например, [7], [8]).

При интерпретации SBH-ADD как проблемы ограниченной сложности в качестве генов рассматриваются элементы множества X. Хромосомы исходной популяции — случайные последовательности генов, содержащие все множество X. Для пары последовательных генов $x_i x_{i+1}$ значение частичной функции приспособленности $f_i(x_i, x_{i+1})$ определяется как длина максимального собственного подслова w слова x_i такого, что для некоторых y_i, y_{i+1} имеют место равенства $x_i = y_i w$, $x_{i+1} = w y_{i+1}$. При таком способе кодирования очевидным образом выполняются все три условия проблемы ограниченной сложности. Следует отметить, что так определенная функция приспособленности минимизирует лишь значение L. Однако поскольку ограничение значения L ограничивает и значение m, генетический алгоритм опосредованно минимизирует и значение параметра m. Выбор точки скрещивания желательно осуществлять не случайным образом, а методом рулетки, основанным на значениях частичных функций приспособленности. Для ускорения сходимости можно на каждом шаге определять несколько точек скрещивания, исходя из значений частичных функций приспособленности. В результате скрещивания хромосома может потерять важное свойство: наличие всех элементов множества X. Поэтому имеет смысл в качестве мутации допускать не только замену, но и удаление и вставку. При этом потеря равенства длин хромосом не имеет существенного значения. В частности, точку скрещивания в паре хромосом можно определять индивидуально. Кроме того, наличие удаления и вставки оправданно с биологической точки зрения.

Список литературы

1. P. A. Pevzner, Y. P. Lysov, K. R. Khrapko, A. V. Belyavsky, V. L. Florentiev, and A. D. Mirzabejov. Improved chips for sequencing by hybridization // Journal of Biomolecular Structure & Dynamics. 1991. V. 9. N 2. P. 399 – 410.

2. P. A. Pevzner. 1-Tuple DNA sequencing: computer analysis // Journal of Biomolecular Structure & Dynamics. 1989. V. 7. N 1. P. 63 – 73.

3. H. L. Bodlaender, R. G. Downey, M. R. Fellows, M. T. Hallett, and H. T. Wareham. Parameterized complexity analysis in computational biology // Computer Applications in the Biosciences. 1995. V. 11. P. 49 – 57.

4. В. Ю. Попов. Вычислительная сложность проблем, связанных с расшифровкой ДНК гибридизацией // Доклады Академии Наук. 2005. Т. 403. N 3. С. 1 – 3.

5. Д. Рутковская, М. Пилиньский, Л. Рутковский. Нейронные сети, генетические алгоритмы и нечеткие системы. М.:"Горячая линия — Телеком", 2004.

6. H. Mühlenbein, T. Mahnig. Convergence theory and applications of the factorized distribution algorithm // Journal of Computing and Information Technology. 1999. V. 7. P. 19 – 32.

7. G. Harik. Leurning linkuge to efficiently solve problems of bounded difficulty using genetic algorithms. PhD thesis, University of Michigan, Ann Arbor, 1997.

8. H. Kargupta. SEARCH, polynomial complexity, and the fast messy genetic algorithm. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

Интеллектуальные системы выбора сценариев экономического роста страны на основе интеллектуального и потенциального капитала нации

Порохня Василий Михайлович,
Проректор-директор института последипломного образования ГУ «ЗИГМУ», проф., д.э.н., д.т.н.,
email:v_prhn@zhu.edu.ua
Колесник Юлия Алексеевна,
email: julia.kolesnik@gmail.com
ст. преп., к.е.н. каф. экономической кибернетики и статистики ГУ «ЗИГМУ», Украина, 69002, г. Запорожье, ул. Жуковского 70б, ГУ «ЗИГМУ»;
Кухарева Лилия Васильевна,
e-mail: l.koukhareva@kres.ru,
Управляющий партнер КРЕС,
127521, г. Москва, ул. Шереметьевская, д. 47, офис 519

Обеспечение стабильности и эффективности функционирования экономики государства в нестабильных рыночных условиях в первую очередь зависит от качества и своевременности принятия управленческого решения о выборе перспективных направлений развития экономики страны. Повышение гибкости и точности управленческих решений возможно только при наличии эффективного инструмента, который позволяет формировать возможные сценарии экономического роста государства, проводить их моделирование и осуществлять оценку привлекательности сценариев с весомым обоснованием последствий выбора каждого из них.

Экономические отношения как на микро- так и на макроуровне требуют использования методов и моделей искусственного интеллекта, которые позволили бы выделять возможные сценарии экономического роста страны на основе накопленных знаний об экономическом объекте в базе знаний. В настоящее время большинство решений в области управления экономическими процессами осуществляется с помощью малоэффективных информационных технологий, не предназначенных для решения сложных интеллектуальных задач.

В данном исследовании осуществлена постановка и предложено решение актуальной задачи построения интеллектуальной системы моделирования макроэкономического роста государства. Для решения поставленной задачи разработан алгоритм моделирования сценариев потенциального экономического роста (ПЭР) государства в зависимости от потенциального и человеческого капитала нации (рис.1 – Моделирование сценариев экономического роста). В основу алгоритма положены разработанные модели оценки интеллектуального капитала (IC) и потенциального капитала (PC), где впервые используется понятие «напряженность экономической среды и потенциальный капитал нации»,которые определяют стратегии потенциального роста экономики(человеческий потенциал).

Данные факторы являются определяющими при исследовании производственной функции, как показателя макроэкономического роста государства, которая существенно отличается от производственных функций экзогенного и эндогенного экономического роста.

Интеллектуальный капитал (IC) выражен следующим образом:

$$IC = HC + CC + SC \tag{1}$$

где HC – человеческий, CC – потребительский , SC – структурный капитал.

Напряженность экономической среды (E) – это способность экономической среды осуществлять технологические преобразования в результате накопления необходимого заряда, уровня интеллектуального капитала для соответствующего преобразования. Напряженность экономической среды определяется следующим образом:

$$E = EVA(t) / \left(1 - \lambda \frac{HC_{2t}}{HC(t)}\right) \tag{2}$$

где

EVA(t) – валовая экономическая добавленная стоимость на технологическом укладе n;

$\lambda>0$ – параметр продуктивности исследовательской технологии в технологическом укладе n;

HC_{2t} – объем труда в исследовательском секторе на технологическом укладе n;

HC(t) – общий объем труда в на технологическом укладе n.

Исходя из уровня напряженности экономической среды потенциальный капитал технологических преобразований выражен как:

$$PC = K_{IC}(t) \cdot E \tag{3}$$

где K_{IC} – коэффициент интеллектуального капитала экономической системы на технологическом укладе в момент времени t.

Предложенный в данной работе метод формирования и выбора сценариев экономического роста включает в себя следующие этапы:

1) выбор числа k кластеров (сценариев) – проводится кластеризация возможных сценариев ЭР на основании нескольких возможных значений k;

2) формирование множества *X* возможных значений структуры экономической системы на основании факторных показателей (IC, PC) и результативного показателя количественной оценки макроэкономического роста (Y);

3) начальное установление центров кластеров – выбор прототипов, которые предполагают существование различных путей достижения стратегических целей государства в зависимости от текущего экономического развития страны и существующих ограничений;

4) нахождение расстояний для установленных центров кластеров с целью выявления сгруппированных точек состояний экономической системы;

5) кластеризация сценариев ЭР на основании минимизации расстояния до центра кластера (рис. 2).

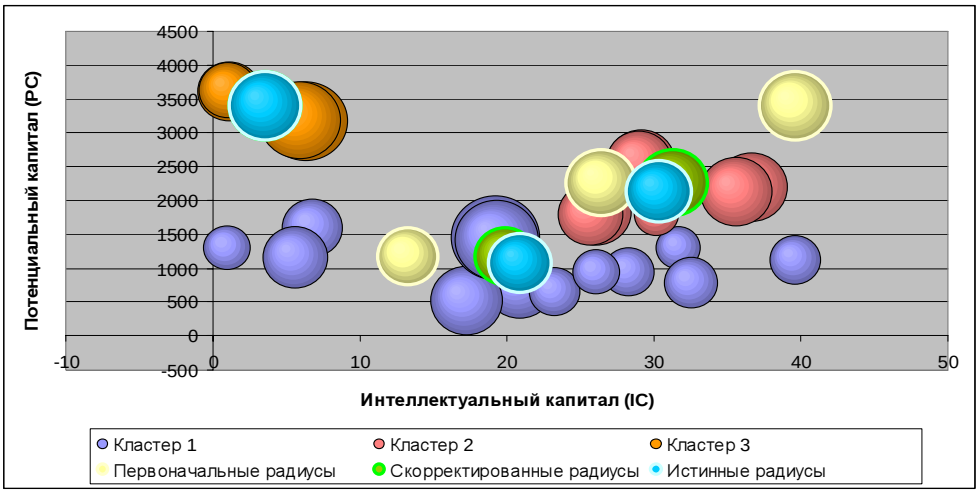


Рис.2 Кластеризация сценариев экономического роста Украины

6) корректирование центров кластеров – на основании полученных множеств состояний экономической системы каждого конкретного кластера;

7) проверка степени стабилизации центров кластеров для выявления момента завершения процесса кластеризации;

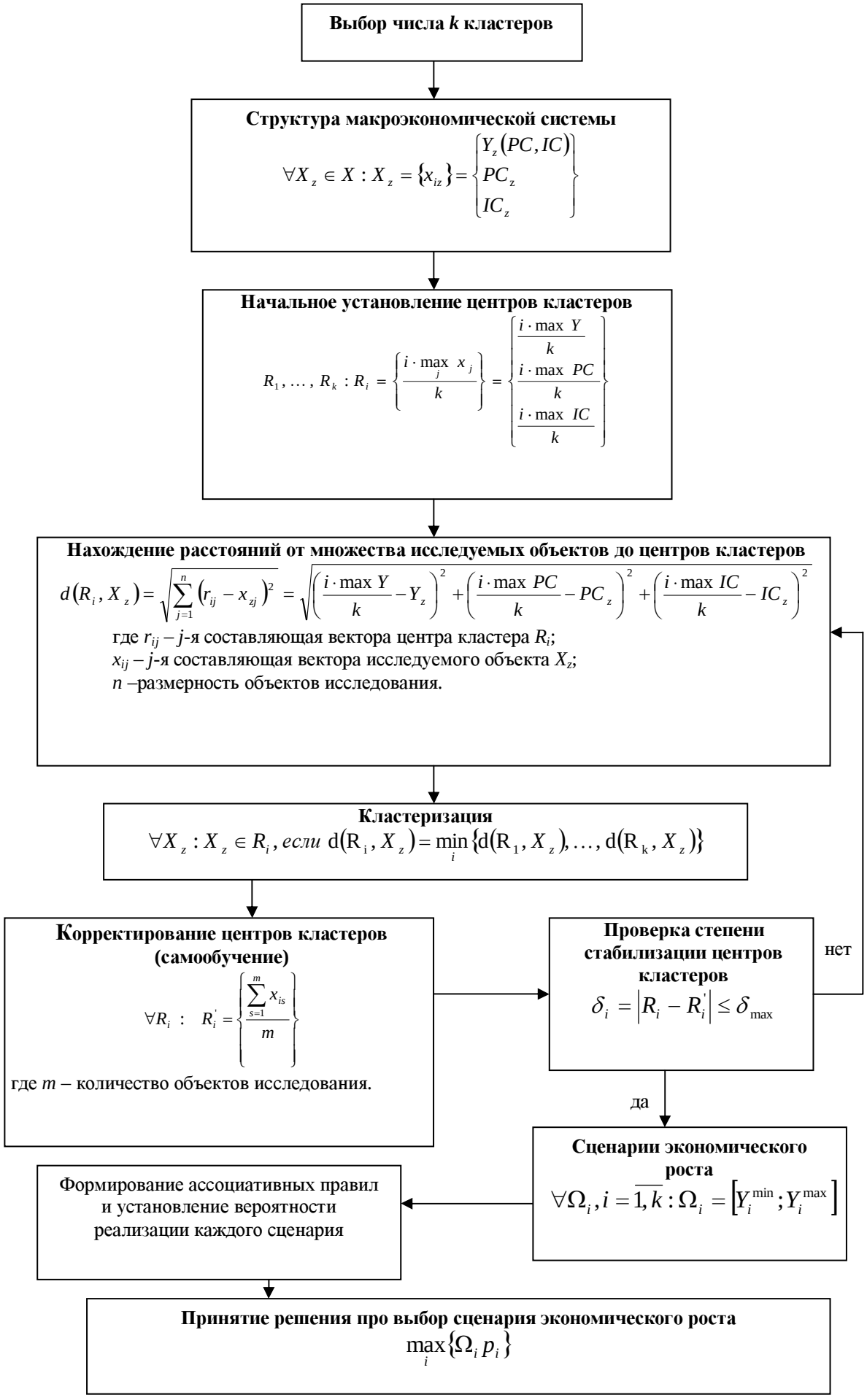
8) формирование сценариев экономического роста на основании полученных фактов;

9) формирование ассоциативных правил и проверка достоверности (уровень поддержки) и установление вероятности реализации каждого сценария при соответствующих условиях:

10) формирование множества *X* возможных значений структуры экономической системы на основании факторных показателей (IC, PC) и результативного показателя количественной оценки макроэкономического роста (Y);

11) начальное установление центров кластеров – выбор прототипов, которые предполагают существование различных путей достижения стратегических целей государства в зависимости от текущего экономического развития страны и существующих ограничений;

12) нахождение расстояний для установленных центров кластеров с целью выявления сгруппированных точек состояний экономической системы;


Выбор числа k кластеров
Структура макроэкономической системы
$\forall X_z \in X : X_z = \{x_{iz}\} = \begin{Bmatrix} Y_z(PC, IC) \\ PC_z \\ IC_z \end{Bmatrix}$
Начальное установление центров кластеров
$R_1, \ldots, R_k : R_i = \left\{ \frac{i \cdot \max\limits_j x_j}{k} \right\} = \begin{Bmatrix} \frac{i \cdot \max Y}{k} \\ \frac{i \cdot \max PC}{k} \\ \frac{i \cdot \max IC}{k} \end{Bmatrix}$
Нахождение расстояний от множества исследуемых объектов до центров кластеров
$d(R_i, X_z) = \sqrt{\sum_{j=1}^{n} (r_{ij} - x_{zj})^2} = \sqrt{\left(\frac{i \cdot \max Y}{k} - Y_z\right)^2 + \left(\frac{i \cdot \max PC}{k} - PC_z\right)^2 + \left(\frac{i \cdot \max IC}{k} - IC_z\right)^2}$
где r_{ij} – j-я составляющая вектора центра кластера R_i;
x_{ij} – j-я составляющая вектора исследуемого объекта X_z;
n –размерность объектов исследования.
Кластеризация
$\forall X_z : X_z \in R_i, \text{если } d(R_i, X_z) = \min_i \{d(R_1, X_z), \ldots, d(R_k, X_z)\}$
Корректирование центров кластеров (самообучение)
$\forall R_i : \quad R_i' = \begin{Bmatrix} \frac{\sum_{s=1}^{m} x_{is}}{m} \end{Bmatrix}$
где m – количество объектов исследования.
Проверка степени стабилизации центров кластеров
$\delta_i = |R_i - R_i'| \le \delta_{\max}$
нет
да
Сценарии экономического роста
$\forall \Omega_i, i = \overline{1,k} : \Omega_i = [Y_i^{\min}; Y_i^{\max}]$
Формирование ассоциативных правил и установление вероятности реализации каждого сценария
Принятие решения про выбор сценария экономического роста
$\max_i \{\Omega_i p_i\}$

13) кластеризация сценариев ЭР на основании минимизации расстояния до центра кластера (рис. 2).

14) корректирование центров кластеров – на основании полученных множеств состояний экономической системы каждого конкретного кластера;

15) проверка степени стабилизации центров кластеров для выявления момента завершения процесса кластеризации;

16) формирование сценариев экономического роста на основании полученных фактов;

17) формирование ассоциативных правил и проверка достоверности (уровень поддержки) и установление вероятности реализации каждого сценария при соответствующих условиях:

Таблица 1: Уровень поддержки сформированных сценариев экономического роста

	Кол-во	**Поддержка (*d*)**	**Экономический рост (нормир. ед.)**	**Ожидаемая эффективность** $\left(d \cdot \overline{ЭР}\right)$
Сценарий 1	13	54,17%	10-25	9,48
Сценарий 2	7	29,17%	25-35	8,75
Сценарий 3	4	16,67%	35-45	6,67

Таблица 2: Ассоциативные правила с вероятностной оценкой сценариев ЭР страны

IC\PC	**0-1000**		**1000-2000**		**2000-3000**		**3000-4000**	
	Ω	p	Ω	p	Ω	p	Ω	p
0-10	-	0,00%	C1	100,00%	C3	100,00%	C1/C2	35% / 65%
10-20	C2	100,00%	C3	100,00%	-	0,00%	-	0,00%
20-30	C1/C2	38% / 62%	C1/C2	35% / 65%	C2	100,00%	-	0,00%
30-40	C1	100,00%	C1	100,00%	C2	100,00%	-	0,00%

18) принятие решения о выборе сценария экономического роста страны на основании максимизации ожидаемой эффективности каждого сценария (рис.3):

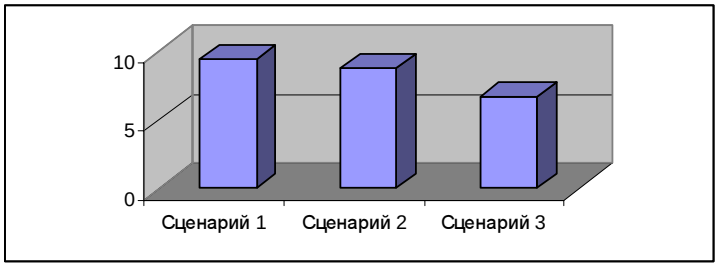


Рис. 3. Ожидаемая эффективность сценариев экономического роста страны

Таким образом, в данном исследовании предложена новая модель экономического роста и новый подход к моделированию сценариев экономического роста страны, базирующийся на использовании методов искусственного интеллекта, который доказывает эффективность использования интеллектуальных систем при решении сложных экономических задач. Практической ценностью предложенных методов является возможность проведения ситуационного анализа стратегий экономического роста с использованием сценарного подхода, как для экономики в целом, так и для отдельного технологического уклада.

Случайная и нечеткая морфология (эмпирическое восстановление модели, идентификация) [1]

Ю. П. Пытьев,
зав. кафедрой КМФ Физического факультета МГУ,
119992, ГСП-2, Москва, Ленинские горы, д. 1, стр. 2,
А. В. Зубюк,
кафедра КМФ Физического факультета МГУ,
адрес: Москва, Ленинградское ш., д. 19, кв. 85,
e-mail: zubuk@cmpd2.phys.msu.su)

Математические методы морфологического анализа изображений разработаны для решения задач анализа и интерпретации реальных сцен на основе их изображений, полученных при неопределенных условиях, таких, например, как характер освещения, его спектральный состав, оптические характеристики объектов и т. п. Источником проблем, возникающих при решении таких задач, является тот факт, что изображение объектов существенно зависит от перечисленных условий в то время как его «видимая на изображении форма» — не зависит. Методы морфологического анализа изображений ориентированы прежде всего на анализ формы изображенных объектов в терминах, инвариантных относительно условий получения изображений.

Пусть моделью изображения является элемент евклидова пространства $\mathcal{R}_N$. Тогда всевозможные изменения условий его регистраций приведут к тому, что изображение одного и того же объекта будет изменяться в пределах некоторого множества V пространства $\mathcal{R}_N$. Это множество называется *формой изображения этого объекта* [1], т. к. оно отражает его характеристики, не зависящие от условий регистрации изображений.

В докладе определены понятия случайной и нечеткой форм изображений и их применения в задачах идентификации формы изображения.

1 Случайная и нечеткая формы изображений

Пусть заданы разбиение $\bigcup\limits_{\omega\in\Omega}\omega=\mathcal{R}_N$ пространства $\mathcal{R}_N$ на непересекающиеся множества $\omega\subset\mathcal{R}_N$, совокупность которых обозначим Ω, σ-алгебра $\mathcal{A}$ подмножеств множества Ω, содержащая все одноточечные множества $\{\omega\}$, $\omega\in\Omega$, и вероятность $\mathrm{Pr}:\mathcal{A}\to[0,1]$. Вероятностное пространство $(\Omega,\mathcal{A},\mathrm{Pr})$, в котором элементарными событиями являются формы $\omega\in\Omega$ в пространстве $\mathcal{R}_N$ является математической моделью случайной формы.

Определение 1. *Случайной формой элементов пространства $\mathcal{R}_N$ назовем вероятностное пространство $(\Omega,\mathcal{A},\mathrm{Pr})$, где Ω — множество непересекающихся форм (подмножеств), образующих разбиение пространства $\mathcal{R}_N$, $\mathcal{A}$ — некоторая σ-алгебра подмножеств множества Ω, а Pr — заданная на ней вероятность.*

Каждому событию $A\in\mathcal{A}$ соответствует форма $V=\bigcup\limits_{\omega\in A}\omega\subset\mathcal{R}_N$, вероятность которой есть $\mathrm{Pr}(A)$. Множество всех форм, соответствующих σ-алгебре $\mathcal{A}$ обозначим $\mathbb{V}_{\mathcal{A}}$, $\mathbb{V}_{\mathcal{A}}=\left\{V:V=\bigcup\limits_{\omega\in A}\omega,\,A\in\mathcal{A}\right\}$.

Пусть, например, множество Ω состоит из всех лучей в $\mathcal{R}_N$, исходящих из начала координат, и S_N — единичная сфера в $\mathcal{R}_N$ с центром в начале координат. Пусть $\mathcal{A}'$ — борелевская σ-алгебра подмножеств сферы S_N, а $\mathcal{A}$ — взаимно однозначно связанная с ней σ-алгебра подмножеств множества Ω. Задав на $\mathcal{A}'$ вероятность Pr, зададим соответствующую вероятность и на $\mathcal{A}$. При этом соотвествующее $\mathcal{A}$ множество форм $\mathbb{V}_{\mathcal{A}}$ содержит все линейные подпространства в $\mathcal{R}_N$, все конусы с вершиной в начале координат и т. п.

По аналогии с определением 1 введем понятие нечеткой формы:

Определение 2. *Нечеткой формой элементов пространства $\mathcal{R}_N$ назовем пространство с возможностью $(\Omega,\mathcal{A},\mathrm{P})$, где Ω — множество непересекающихся форм, образующих разбиение $\mathcal{R}_N$, $\mathcal{A}$ — некоторая σ-алгебра подмножеств Ω, а P — заданная на ней возможность.*

[1] Работа выполнена при финансовой поддержке РФФИ грант № 05-01-00532-а

2 Гранулирование пространства элементарных событий

Задача гранулирования множества элементарных событий возникает при построении возможности, имеющей стохастический прототип, т. е. в том случае, когда заданной теоретико-вероятностной модели $(\Omega, \mathcal{A}, \Pr)$ требуется сопоставить теоретико-возможностную модель $(\Omega, \mathcal{A}, \mathrm{P})$. Для этого возможность P задается таким образом, чтобы для любых двух событий $A, B \in \mathcal{A}$ из соотношения $\Pr(A) \leqslant \Pr(B)$ следовало соотношение $\mathrm{P}(A) \leqslant \mathrm{P}(B)$. Любая такая возможность P называется согласованной с вероятностью Pr. Однако в ряде случаев, в том числе в случае абсолютно непрерывной вероятности Pr, всякая согласованная с ней возможность оказывается «тривиальной», т. е. для любого $A \in \mathcal{A} \quad \mathrm{P}(A) = 1$. В таких ситуациях для того, чтобы построенная возможность отражала особенности исходной вероятности, множество элементарных событий Ω следует предварительно гранулировать, т. е. разбить на измеримые множества $\Omega_1, \Omega_2, \ldots$, которые и считать в дальнейшем элементарными событиями. При этом гранулирование следует выполнить таким образом, чтобы при последующем построении возможности, согласованной с сужением вероятности Pr на некоторую σ-алгебру подмножеств множества $\{\Omega_1, \Omega_2, \ldots\}$ возможности новых элементарных событий оказались различными. Этому требованию отвечают разбиения из определения 3.

Пусть задано множество элементарных событий Ω и абсолюно непрерывная вероятность Pr, заданная плотностью $\mathrm{pr}(\cdot) : \Omega \to [0, \infty)$, $\int\limits_{\Omega} \mathrm{pr}(\omega)\mathrm{d}\omega = 1$.

Определение 3 (Почти оптимальное гранулирование[2]. */ Почти оптимальным разбиением множества Ω на m гранул назовем любое измеримое разбиение $\Omega_1 \cup \ldots \cup \Omega_m = \Omega$, для которого*

$$\Omega_i = \{\omega : g_i \leqslant \mathrm{pr}(\omega) < g_{i-1}\}, \quad \Pr(\Omega_i) > \frac{1}{2^i}, \quad i = 1, \ldots, m,$$
$$g_0 = +\infty, \quad g_m = 0.$$

Заметим, что для любого почти оптимального разбиения Ω на m гранул $\Omega_1, \ldots, \Omega_m \quad \frac{1}{2^i} < \Pr(\Omega_i) < \frac{1}{2^i} + \frac{1}{2^m}$, $i = 1, \ldots, m$.

Определение 4 (Почти оптимальное гранулирование с помощью кластеров). *Пусть задано разбиение множества Ω на n кластеров $\mathcal{C}l_1 \cup \ldots \cup \mathcal{C}l_n = \Omega$. Почти оптимальным разбиением множества Ω на m гранул с помощью кластеров $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$ назовем любое разбиение $\Omega'_1 \cup \ldots \cup \Omega'_m = \Omega$, для которого*

$$J'_i = \left\{ j : g'_i \leqslant \frac{\Pr(\mathcal{C}l_j)}{v(\mathcal{C}l_j)} < g'_{i-1} \right\}, \quad \Omega'_i = \bigcup_{j \in J'_i} \mathcal{C}l_j, \quad \Pr(\Omega'_i) > \frac{1}{2^i},$$
$$i = 1, \ldots, m, \quad g'_0 = +\infty, \quad g'_m = 0,$$

где $v(\mathcal{C}l_j) \stackrel{def}{=} \int\limits_{\mathcal{C}l_j} \mathrm{d}\omega$, $j = 1, \ldots, n$.

Следующая лемма устанавливает связь между почти оптимальным гранулированием и почти оптимальным гранулированием с помощью кластеров.

Лемма 1. *Если плотность вероятности* $\mathrm{pr}(\cdot)$ *и кластеры* $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$ *таковы, что*

1. *функция* $F(g) \stackrel{def}{=} \int\limits_{g \leqslant \mathrm{pr}(\omega)} \mathrm{pr}(\omega)\mathrm{d}\omega$, $g \in [0, \infty)$ *непрерывна,*
2. *для любых двух кластеров* $\mathcal{C}l_i$ *и* $\mathcal{C}l_j$, $i \neq j$ *из соотношения* $\frac{\Pr(\mathcal{C}l_i)}{v(\mathcal{C}l_i)} \leqslant \frac{\Pr(\mathcal{C}l_j)}{v(\mathcal{C}l_j)}$ *следуют соотношения* $\forall \omega \in \mathcal{C}l_i \quad \mathrm{pr}(\omega) \leqslant \frac{\Pr(\mathcal{C}l_j)}{v(\mathcal{C}l_j)}, \quad \forall \omega \in \mathcal{C}l_j \quad \frac{\Pr(\mathcal{C}l_i)}{v(\mathcal{C}l_i)} \leqslant \mathrm{pr}(\omega),$

то для любого почти оптимального разбиения Ω на m гранул $\Omega'_1, \ldots, \Omega'_m$ с помощью кластеров $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$ найдется такое почти оптимальное разбиение Ω на m гранул $\Omega_1, \ldots, \Omega_m$, что для любого $i = 1, \ldots, m$ будет выполнено равенство $\Pr(\Omega_i) = \Pr(\Omega'_i)$, и любой кластер $\mathcal{C}l_j \subset \Omega'_i$ будет содержать хотя бы одну точку соответствующей гранулы Ω_i, $i = 1, \ldots, m$.

3 Алгоритм эмпирического гранулирования множества элементарных событий с помощью кластеров

Пусть дано разбиение Ω на n кластеров $\mathcal{C}l_1 \cup \ldots \cup \mathcal{C}l_n = \Omega$. Пусть доступны наблюдению исходы взаимно независимых стохастических экспериментов, контролируемых вероятностью с плотностью $\mathrm{pr}(\cdot) : \Omega \to [0, \infty)$, $\int\limits_{\Omega} \mathrm{pr}(\omega)\mathrm{d}\omega = 1$. Пусть плотность $\mathrm{pr}(\cdot)$ и кластеры $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$ таковы, что существует почти оптимальное разбиение Ω на m гранул с помощью кластеров $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$. Требуется по наблюдениям ω_1, ω_2, ... экспериментов построить это разбиение.

Для этого зафиксируем некоторую величину $\alpha \in [0, 1)$. Определим $\delta_l(\alpha) \stackrel{\mathrm{def}}{=} \sqrt{\frac{\ln \frac{n+m}{1-\alpha}}{2l}}$, $l = 1, 2, \ldots$ Обозначим $\nu_l(A)$ частоту события $A \subset \Omega$ в серии l экспериментов, т. е. $\nu_l(A) \stackrel{\mathrm{def}}{=} \frac{1}{l}\sum_{i=1}^{l} \chi_A(\omega_i)$, где $\chi_A(\omega) = 1$, если $\omega \in A$, $\chi_A(\omega) = 0$ в остальных случаях, и

$$\begin{aligned} &\widetilde{J}_l(g_1, g_2) \stackrel{\mathrm{def}}{=} \left\{ j : g_2 \leqslant \frac{\nu_l(\mathcal{C}l_j) - \delta_l(\alpha)}{v(\mathcal{C}l_j)} < \frac{\nu_l(\mathcal{C}l_j) + \delta_l(\alpha)}{v(\mathcal{C}l_j)} < g_1 \right\}, \\ &\widetilde{\Omega}_l(g_1, g_2) \stackrel{\mathrm{def}}{=} \bigcup_{j \in \widetilde{J}_l(g_1, g_2)} \mathcal{C}l_j. \end{aligned} \tag{1}$$

Будем проводить эксперименты до тех пор, пока не найдутся такие $+\infty = \widetilde{g}_0 > \widetilde{g}_1 > \ldots > \widetilde{g}_m = 0$, при которых выполнятся условия

$$\nu_l\left(\widetilde{\Omega}_l(\widetilde{g}_{i-1}, \widetilde{g}_i)\right) - \delta_l(\alpha) > \frac{1}{2^i}, \quad i = 1, \ldots, m. \tag{2}$$

При выполнении условий (2) разбиение $\widetilde{\Omega}_1 \cup \ldots \cup \widetilde{\Omega}_m = \Omega$, где $\widetilde{\Omega}_i \stackrel{\mathrm{def}}{=} \widetilde{\Omega}_l(\widetilde{g}_{i-1}, \widetilde{g}_i)$, $i = 1, \ldots, m$, с вероятностью не менее α будет почти оптимальным с использованием кластеров $\mathcal{C}l_1, \ldots, \mathcal{C}l_n$, причем вероятность того, что алгоритм завершится на l-ом шаге, стремится к единице при $l \to \infty$.

Описанный алгоритм позволяет, в частности, эмпирически восстанавливать нечеткую форму изображения (элемента евклидова пространства $\mathcal{R}_N$), имеющую стохастический прототип, [2, 3].

4 Идентификация изображений, имеющих случайную форму

Пусть заданы две случайные формы: $F_1 = (\Omega, \mathcal{A}, \mathrm{Pr}_1)$ и $F_2 = (\Omega, \mathcal{A}, \mathrm{Pr}_2)$, где вероятности Pr_1 и Pr_2 заданы плотностями $\mathrm{pr}^{(1)}(\cdot)$ и $\mathrm{pr}^{(2)}(\cdot)$ соответственно, и предъявляемое для идентификации изображение ξ формируется по схеме

$$\xi = f + \nu, \tag{3}$$

где $f \in \mathcal{R}_N$ — произвольный элемент случайной формы F_1 или F_2, а ν — случайный элемент пространства $\mathcal{R}_N$ с плотностью распределения $\mathrm{pr}_\nu(\cdot)$ (случайные элементы $\omega \in \Omega$ и $\nu \in \mathcal{R}_N$ независимы). Требуется по предъявленному изображению ξ определить, какую случайную форму (F_1 или F_2) имеет элемент f.

Заметим, что если f имеет случайную форму $F_t, t = 1, 2$, то ξ является случайным элементом пространства $\mathcal{R}_N$ с одним из следующих плотностей рапределения: $\mathrm{pr}_\xi^{(t,\,\phi)}(x) = \int\limits_{\Omega} \mathrm{pr}^{(t)}(\omega)\,\mathrm{pr}_\nu(x - \phi_\omega)\,\mathrm{d}\omega$, где символом ϕ обозначена произвольная совокупность элементов $\mathcal{R}_N$ вида $\phi = \{\phi_\omega \in \omega, \omega \in \Omega\}$.

Введем следующие множества распределений:

$$\mathbb{P}\mathrm{r}_1 \stackrel{\mathrm{def}}{=} \left\{\mathrm{pr}_\xi^{(1,\,\phi)}\right\}, \mathbb{P}\mathrm{r}_2 \stackrel{\mathrm{def}}{=} \left\{\mathrm{pr}_\xi^{(2,\,\phi)}\right\}, \mathcal{H}_0 \stackrel{\mathrm{def}}{=} \mathbb{P}\mathrm{r}_1 \cap \mathbb{P}\mathrm{r}_2, \mathcal{H}_t \stackrel{\mathrm{def}}{=} \mathbb{P}\mathrm{r}_t \setminus \mathcal{H}_0, t = 1, 2.$$

Для идентификации предъявленного изображения можно воспользоваться минимаксным критерием $(\widetilde{\pi}_0(x), \widetilde{\pi}_1(x), \widetilde{\pi}_2(x))$, $x \in \mathcal{R}_N$ проверки гипотез $\mathcal{H}_0$, $\mathcal{H}_1$, $\mathcal{H}_2$, который минимизирует вероятность ошибки и находится как решение задачи на минимакс [2]

$$\begin{cases} \max(\alpha_0, \alpha_1, \alpha_2) \sim \min\limits_{\pi_0, \pi_1, \pi_2}, \\ \pi_0(x) + \pi_1(x) + \pi_2(x) = 1,\ x \in \mathcal{R}_N \\ \pi_i(x) \geqslant 0,\ x \in \mathcal{R}_N,\ i = 0, 1, 2, \end{cases} \tag{4}$$

где $\alpha_i \stackrel{\text{def}}{=} \max\limits_{\mathrm{pr}_\xi^{(t,\,\phi)}(\cdot) \notin \mathcal{H}_i} \int\limits_{\mathcal{R}_N} \mathrm{pr}_\xi^{(t,\,\phi)}(x)\, \pi_i(x)\, \mathrm{d}x,\ i = 0, 1, 2.$

При этом гипотеза $\mathcal{H}_1$ свидетельствует в пользу случайной формы F_1, гипотеза $\mathcal{H}_2$ — в пользу F_2, а гипотеза $\mathcal{H}_0$ означает, что в рамках постановки (5) по предъявленному изображению невозможно определить, изображением какой случайной формы является f.

5 Идентификация изображений, имеющих нечеткую форму

Пусть заданы две нечеткие формы: $F_1 = (\Omega, \mathcal{A}, \mathrm{P}_1)$ и $F_2 = (\Omega, \mathcal{A}, \mathrm{P}_2)$, где возможности P_1 и P_2 заданы распределениями $\mathrm{p}^{(1)}(\cdot)$ и $\mathrm{p}^{(2)}(\cdot)$ соответственно, и предъявляемое для идентификации изображение ξ формируется по схеме (3), где $f \in \mathcal{R}_N$ — произвольный элемент нечеткой формы F_1 или F_2, а ν — нечеткий элемент пространства $\mathcal{R}_N$ с распределением возможности $\mathrm{p}_\nu(\cdot)$ (нечеткие элементы $\omega \in \Omega$ и $\nu \in \mathcal{R}_N$ независимы). Требуется по предъявленному изображению ξ определить, какую нечеткую форму (F_1 или F_2) имеет элемент f.

Заметим, что если f имеет нечеткую форму F_t, $t = 1, 2$, то ξ является нечетким элементом пространства $\mathcal{R}_N$ с одним из следующих рапределений возможности: $\mathrm{p}_\xi^{(t,\,\phi)}(x) = \max\limits_{\omega \in \Omega} \min\Big(\mathrm{p}^{(t)}(\omega),\, \mathrm{p}_\nu(x - \phi_\omega)\Big)$, где символом ϕ обозначена произвольная совокупность элементов $\mathcal{R}_N$ вида $\phi = \{\phi_\omega \in \omega,\, \omega \in \Omega\}$.

Введем следующие множества распределений:

$$\mathbb{P}_1 \stackrel{\text{def}}{=} \left\{\mathrm{p}_\xi^{(1,\,\phi)}\right\},\ \mathbb{P}_2 \stackrel{\text{def}}{=} \left\{\mathrm{p}_\xi^{(2,\,\phi)}\right\},\quad \mathcal{H}_0 \stackrel{\text{def}}{=} \mathbb{P}_1 \cap \mathbb{P}_2,\ \mathcal{H}_1 \stackrel{\text{def}}{=} \mathbb{P}_1 \setminus \mathcal{H}_0,\ \mathcal{H}_2 \stackrel{\text{def}}{=} \mathbb{P}_2 \setminus \mathcal{H}_0.$$

Для идентификации предъявленного изображения можно воспользоваться минимаксным критерием $(\widetilde{\pi}_0(x), \widetilde{\pi}_1(x), \widetilde{\pi}_2(x))$, $x \in \mathcal{R}_N$ проверки гипотез $\mathcal{H}_0$, $\mathcal{H}_1$, $\mathcal{H}_2$, который минимизирует возможность ошибки и находится как решение задачи на минимакс [2]

$$\begin{cases} \max(\alpha_0, \alpha_1, \alpha_2) \sim \min\limits_{\pi_0, \pi_1, \pi_2}, \\ \max(\pi_0(x),\, \pi_1(x),\, \pi_2(x)) = 1,\ x \in \mathcal{R}_N \\ \pi_i(x) \geqslant 0,\ x \in \mathcal{R}_N,\ i = 0, 1, 2, \end{cases} \tag{5}$$

где $\alpha_i \stackrel{\text{def}}{=} \max\limits_{\mathrm{p}_\xi^{(t,\,\phi)}(\cdot) \notin \mathcal{H}_i} \sup\limits_{x \in \mathcal{R}_N} \min\Big(\mathrm{p}_\xi^{(t,\,\phi)}(x),\, \pi_i(x)\Big),\ i = 0, 1, 2.$

При этом принятие той или иной гипотезы интерпретируется аналогично пункту 4.

Одним из решений задачи (5) является правило идентификации

$$\pi_i(x) = \begin{cases} 1, \text{ если } \min\limits_j \max\limits_{\mathrm{p}_\xi^{(t,\,\phi)}(\cdot) \notin \mathcal{H}_j} \mathrm{p}_\xi^{(t,\,\phi)}(x) = \max\limits_{\mathrm{p}_\xi^{(t,\,\phi)}(\cdot) \notin \mathcal{H}_i} \mathrm{p}_\xi^{(t,\,\phi)}(x), \\ 0, \text{ в остальных случаях.} \end{cases} \qquad i = 0, 1, 2$$

Список литературы

1. Пытьев Ю. П., Морфологический анализ изображений, Докл. АН СССР, 1983, т. 269, № 5, с. 1061-1064

2. Пытьев Ю. П., Возможность как альтернатива вероятности. Математические и эмпирические основы, применения, Физматлит, 2006

3. Зубюк А. В., Эмпирическое восстановление возможности, в сб. докл. XII всероссийской конф. «Математические методы распознавания образов», 2005, с. 112-115

Неопределенные нечеткие модели и их применения [1]

Ю. П. Пытьев
зав. кафедрой КМФ Физического факультета МГУ,
О. В. Фаломкина,
научный сотрудник кафедры КМФ Физического факультета МГУ
адрес: 119992, ГСП-2, Москва, Ленинские горы,
МГУ им. Ломоносова, д. 1, стр. 2, Физический ф-т

Для последних нескольких десятилетий характерен возрастающий интерес к математическим моделям неясности, неопределеноти и т.п., характеризующим неполноту знаний, их недостоверность, и — нечеткости, случайности, и т.п., относящихся к их содержанию.

Как известно, основным инструментом для моделирования нечеткости и неопределенности является теория вероятностей. Как правило нечеткость моделируется распределением вероятностей, а неопределенность — частичным незнанием этого распределения. Однако на практике теория вероятностей как математическая основа моделирования нечеткости и неопределенности часто оказывается неэффективной в связи с невероятностной природой последних, но и в тех случаях, когда стохастический характер нечеткости и неопределенности очевиден, принципиальные трудности, как правило, возникают при эмпирическом построении вероятности [1]. Причины этих трудностей заключаются в том, что для эмпирического построения теоретико-вероятностной модели, равно как и для ее верификации, требуются большие объемы наблюдений, для их получения обычно требуется время, в течение которого вероятностные характеристики объекта эволюционируют, т.е нарушается «принцип устойчивости частот». В этом случае наблюдения за частотами не позволяют оценить вероятность, а знание самих вероятностей не позволяет предсказать частоту, т.е. эмпирически нельзя восстановить теоретико-вероятностную модель[2].

С этим связано появление ряда фундаментальных математических работ, посвященных невероятностым методам моделирования нечеткости, случайности и неясности. Субъективная вероятность Сэведжа как мера неуверенности субъекта, суждения которого удовлетворяют определенным условиям «рациональности»; верхние и нижние вероятности Демпстера, характеризующие неполноту наблюдений и отражающие неопределенность в теории вероятностей, моделируемую многозначными отображениями; правдоподобие и доверие Шеффера, обобщающие конструкции Демпстера в теории принятия решений; возможность Заде, основанная на его теории нечетких множеств, — далеко не полный перечень таких работ. Отметим также фундаментальные результаты, относящиеся к моделированию неопределенности, обусловленной неполнотой знаний. Это мера правдоподобия Фридмана и Халперна, обобщающая доверие Демпстера-Шеффера, возможность Заде и нечеткая мера Сугено, верхнее и нижнее предвидения Вэлли, обобщающие вероятность, возможность Заде и доверие Демпстера-Шеффера, контекстная модель Гебхардта и Круза, использующаяся для описания нечеткости и неопределенности, позволившая дать альтернативную формулировку теории Демстера -Шеффера, и, наконец, мера правдоподобия (доверия) возможности (необходимости) в [2].

Среди упомянутых публикаций нет работ, кроме [2], в которых методы моделирования содержали бы математические средства для формального выражения как мнений исследователя по поводу адекватности используемой модели и основанных на ней выводов, так и эволюции этих мнений, обусловленной получением новых данных. В разработанных в [2] неопределенных нечетких (НН) моделях нечеткость, неточность формулировок, относящаяся к содержанию информации, охарактеризована в терминах значений мер возможности и (или) необходимости, а достоверность формулировок, истинность которых не может быть абсолютной в силу принципиальной неполноты знаний, охарактеризована в терминах значений мер правдоподобия и (или) доверия.

Принцип построения НН модели в [2] состоит в следующем: рассматривается класс теоретико-возможностных моделей, на котором задается распределение правдоподобий, что позволяет разработать правила принятия решений, в которых критерии оптимальности основаны на значениях правдоподобия (доверия) возможности и (или) необходимости ошибки решения, при этом возможность и (или) необходимость определяют содержательную характеристику качества решения, тогда

[1]Работа выполнена при финансовой поддержке РФФИ грант № 05-01-00532-а.

[2]Разумеется, не трудно привести примеры стохастических объектов, модели которых могут быть априори охарактеризованы математически, а эмпирически лишь уточнены и верифицированы. К этой категории, прежде всего, относятся сложные стохастические объекты, эволюция моделей которых хорошо описывается, например, линейными разностными, дифференциальными или интегральными стохастическими уравнениями. Математические модели таких объектов могут быть эмпирически восстановлены на основе временны́х рядов наблюдений за ними, и эти же данные позволяют проверять их адекватность.

как правдоподобие тех или иных значений возможности (необходимости) ошибки, показывающее, в какой степени им следует доверять, является дополнительной характеристикой качества.

Остановимся кратко на формализме неопределенной нечеткой математики. Пусть $(Y, \mathcal{P}(Y), \mathrm{P}^{\eta})$ — пространство с возможностью, которое состоит из множества Y элементарных событий, σ-алгебры $\mathcal{P}(Y)$ всех подмножеств Y, называемых событиями, и функции $\mathrm{P}^{\eta}(\cdot) : \mathcal{P}(Y) \to [0,\,1]$, называемой мерой возможности (возможностью).

Возможность $\mathrm{P}^{\eta}(\cdot)$ определяется ее значениями $f^{\eta}(y) \triangleq \mathrm{P}^{\eta}(\{y\}) = \mathrm{P}^{\eta}(\eta = y),\ y \in Y$, на одноточечных подмножествах $\{y\} \subset Y$, а именно, $\mathrm{P}^{\eta}(A) \triangleq \mathrm{P}^{\eta}(\eta \in A) = \sup\limits_{y \in A} f^{\eta}(y),\ A \in \mathcal{P}(Y)$.

Функция $f^{\eta}(\cdot) : Y \to [0, 1]$ называется распределением возможностей значений канонического для $(Y, \mathcal{P}(Y), \mathrm{P}^{\eta})$ нечеткого элемента $\eta : Y \to (Y, \mathcal{P}(Y), \mathrm{P}^{\eta}(\cdot))$ и распределением возможности $\mathrm{P}^{\eta}(\cdot)$.

Пусть аналогично $(\mathcal{U}, \mathcal{P}(\mathcal{U}), \mathrm{P}l^{\widetilde{u}}(\cdot))$ — *пространство с правдоподобием*, состоящее из множества $\mathcal{U}$ элементарных высказываний, σ-алгебры $\mathcal{P}(\mathcal{U})$ всех подмножеств (высказываний), и функции $\mathrm{P}l^{\widetilde{u}}(\cdot) : \mathcal{P}(\mathcal{U}) \to [0,\,1]$, называемой мерой правдоподобия. Аналогично возможности $\mathrm{P}^{\eta}(\cdot)$, правдоподобие $\mathrm{P}l^{\widetilde{u}}(\cdot)$ определяется распределением правдоподобий $g^{\widetilde{u}}(\cdot) : \mathcal{U} \to [0,\,1]$ значений канонического неопределенного элемента $\widetilde{u} : (\mathcal{U}, \mathcal{P}(\mathcal{U}), \mathrm{P}l^{\widetilde{u}}(\cdot)) \to \mathcal{U}$.

Определение 0.0.1 (2). НН элементом, принимающим значения в X, называется образ $\widetilde{\xi} \triangleq q(\eta, \widetilde{u})$ (упорядоченной) пары $(\eta, \widetilde{u})$ — нечеткого η и неопределенного $\widetilde{u}$ элементов при отображении $q(\cdot, \cdot) : Y \times \mathcal{U} \to X$.

Функция $\tau_x^{\widetilde{\xi}}(p) \triangleq \mathrm{P}l(P(\widetilde{\xi} = x) = p) = \sup\left\{g^{\widetilde{u}}(u) \;\middle|\; u \in \mathcal{U},\ f^{\xi_u}(x) = p\right\},\ x \in X,\ p \in [0,\,1]$, называется распределением правдоподобия возможностей значений НН элемента $\widetilde{\xi}$, или короче - распределением $\widetilde{\xi}$. Ее значение $\tau_x^{\widetilde{\xi}}(p)$ определяет правдоподобие истинности «элементарного» высказывания, согласно которому p — возможность равенства $\widetilde{\xi} = x \in X$.

Определение 0.0.2 (2). Неопределенным нечетким (НН) множеством называется образ $\widetilde{A} = Q(\eta,\, \widetilde{u})$ нечеткого η и неопределенного $\widetilde{u}$ элементов, канонических соответственно для $(Y,\, \mathcal{P}(Y),\, P^{\eta})$ и $(\mathcal{U},\, \mathcal{P}(\mathcal{U}),\, \mathrm{P}l^{\widetilde{u}})$, при многозначном отображении $Q(\cdot,\, \cdot) :\ Y \times \mathcal{U} \to \mathcal{P}(X)$.

Пример решения задачи анализа и интерпретации данных для НН модели. Пусть

$$\widetilde{\xi} = A\widetilde{\varphi} + \widetilde{\nu}, \tag{1}$$

— искаженный шумом $\widetilde{\nu} \in \mathcal{R}_n$ выходной сигнал $A\widetilde{\varphi}$ прибора A, на вход которого поступил сигнал $\widetilde{\varphi} \in \mathcal{F},\ A : \mathcal{R}_m \to \mathcal{R}_n$ — заданный оператор, моделирующий измерительный прибор, $\mathcal{F} \subset \mathcal{R}_m$ — множество, априори содержащее $\widetilde{\varphi}$, $\mathcal{R}_m,\ \mathcal{R}_n$ — конечномерные евклидовы пространства, $m,\ n$ — их размерности. В задаче интерпретации измерения (1) требуется определить правило (интерпретации) $d(\cdot) : \mathcal{R}_n \to \mathcal{U}$ так, чтобы элемент $d(\widetilde{\xi})$ можно было считать наиболее оптимальной оценкой значения $U\widetilde{\varphi}$ параметра исследуемого объекта. Оператор $U : \mathcal{R}_m \to \mathcal{U}$ моделирует идеальный измерительный прибор.

В НН модели $[A, \tau_{\cdot}^{\widetilde{\varphi}}(\cdot), \tau_{\cdot}^{\widetilde{\nu}}(\cdot)]$ схемы измерения (1) ошибка измерения (шум) $\widetilde{\nu}$ — НН вектор $\mathcal{R}_n$, входной сигнал $\widetilde{\varphi}$ — НН вектор $\mathcal{R}_m$, $\tau_{\cdot}^{\widetilde{\nu}}(\cdot) : \mathcal{R}_n \times [0,\,1] \to [0,\,1]$, $\tau_{\cdot}^{\widetilde{\varphi}}(\cdot) : \mathcal{F} \times [0,\,1] \to [0,\,1]$ — их распределения; выходной сигнал $\widetilde{\xi}$ — НН элемент, его распределение $\tau_x^{\widetilde{\xi}}(p) = \tau_{x-Af}^{\widetilde{\nu}}(p),\ x \in \mathcal{R}_n$, зависит от значений неизвестного входного сигнала $\widetilde{\varphi}$ как от параметра. Наконец, параметр $\widetilde{\eta}$ исследуемого объекта — НН элемент $\mathcal{U}$ [2].

Качество НН интерпретации охарактеризовано значениями

$$\mathrm{P}l(N(d(\cdot)) = n) \triangleq \left(\bigvee_{\substack{f \in \mathcal{F},\\ x \in \mathcal{R}_n}} \widehat{\tau}_f^{(d(x))}\right)(\theta(n)), \quad \theta(n) \in [0,\,1],$$

правдоподобия необходимости n ошибки интерпретации, обусловленной использованием правила $d(\cdot)$, где $\theta(\cdot) : [0,\,1] \to [0,\,1]$ — непрерывная, строго монотонная функция, $\theta(0) = 1,\ \theta(1) = 0,\ a \in [0,\,1]$,

$$\widehat{\tau}_f^{(d(x))}(q) =$$

$$= \sup\{\min((\tau_{x-Af}^{\widetilde{\nu}} \wedge \tau_f^{\widetilde{\varphi}})(a),\, \tau_{Uf}^{\mathcal{U}\setminus\widetilde{A}_{(d(x))}}(b))|\, a,\, b \in [0,\,1],\ \min(a,\, b) = q\} \triangleq$$

$$\triangleq (\tau_{x-Af}^{\widetilde{\nu}} \wedge \tau_f^{\widetilde{\varphi}} \wedge \tau_{Uf}^{\mathcal{U}\setminus\widetilde{A}_{(d(x))}})(q), \quad q \in [0,\,1],$$

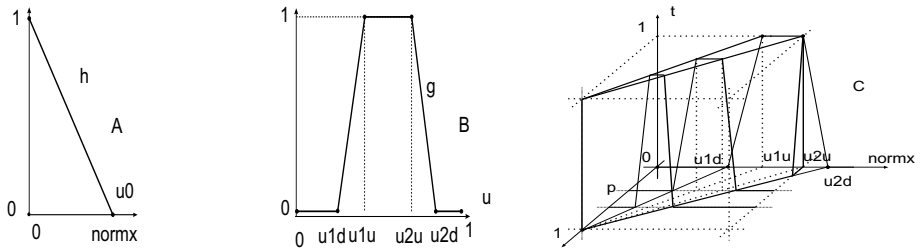


Рис. 1:

$\tau_{\cdot}^{\mathcal{U}\setminus\widetilde{A}_{(d(x))}}(\cdot) : \mathcal{U}\times[0,\,1]\to[0,\,1]$ — индикаторная функция НН множества $\mathcal{U}\setminus\widetilde{A}_{(d(x))}$, $d(\cdot):\mathcal{R}_n\to\mathcal{U},\ x\in\mathcal{R}_n$. Значение $\tau_{Uf}^{\mathcal{U}\setminus\widetilde{A}_{(d(x))}}(p)$ определяет правдоподобие возможности p события $Uf\notin\widetilde{A}_{(d(x))}$. *Правдоподобие возможности ошибки использования значения $d\in\mathcal{U}$ вместо $u\in\mathcal{U}$ определяется как правдоподобие возможности покрытия $u\in\mathcal{U}$ НН множеством $\widetilde{A}_{(d)}$, $d\in\mathcal{U}$.*

НН модель ошибки в (1) задана распределением нечеткого элемента $\nu_u=\widetilde{\nu}|_{\widetilde{u}=u}$ для каждого значения $u\in R_+$ неопределенного элемента $\widetilde{u}$ равенством

$$f^{\nu_u}(x)=h(\|\Sigma^{-1/2}x\|^2,\,u)=\begin{cases}1-\dfrac{\|\Sigma^{-1/2}x\|^2}{u}, & \text{если } 0\leqslant\|\Sigma^{-1/2}x\|^2\leqslant u,\\ 0, & \text{если } \|\Sigma^{-1/2}x\|^2>u,\end{cases}$$

$$\text{где } u>0,\ x\in\mathcal{R}_n;$$
$$f^{\nu_0}(0)=h(0,0)=\quad 1,$$

см.рис. 1(а), где $\Sigma:\mathcal{R}_n\to\mathcal{R}_n$ — положительно определенный оператор. Распределение $g^{\widetilde{u}}(\cdot)$ представлено на рис. 1(б), распределение $\tau_{\cdot}^{\widetilde{\nu}}(\cdot)$ определяется выражением [2] (см. рис. 1(в))

$$\tau_x^{\widetilde{\nu}}(p)=\sup\{g^{\widetilde{u}}(u)|\ u\in R_+,\ h(\|\Sigma^{-1/2}x\|^2,\,u)=p\}=$$
$$=\begin{cases}g^{\widetilde{u}}\left(\frac{\|\Sigma^{-1/2}x\|^2}{1-p}\right)>0, & \text{если } p\in[0,1),\ \|\Sigma^{-1/2}x\|^2/(1-p)\in(\underline{u}_1,\underline{u}_2),\\ 0, & \text{если} p\in[0,1),\ \begin{cases}\|\Sigma^{-1/2}x\|^2/(1-p)\leqslant\underline{u}_1\\ \underline{u}_2\leqslant\|\Sigma^{-1/2}x\|^2/(1-p),\end{cases}\\ 1, & \text{если} p=1,\ \|\Sigma^{-1/2}x\|=0,\quad x\in\mathcal{R}_n,\\ 0, & \text{если} p=1,\ \|\Sigma^{-1/2}x\|>0.\end{cases}\qquad(2)$$

НН модель входного сигнала $\widetilde{\varphi}$ задана распределением $\tau_{\cdot}^{\widetilde{\varphi}}(\cdot)$ аналогично тому, как определена модель $\widetilde{\nu}$ (с заменами $\|\Sigma^{-1/2}x\|^2\to\|F^{-1/2}(f-f_0)\|^2,\ u\to v,\ u_1\to v_1,\ u_2\to v_2$, где $F:\mathcal{F}\to\mathcal{F}$ — положительно определенный оператор, $f_0\in\mathcal{F}$).

Если о входном сигнале априори ничего не известно, то

$$\tau_f^{\widetilde{\varphi}}(p)=\begin{cases}1, & \text{если } p=1,\ f\in\mathcal{F},\\ 0, & \text{если } 0\leqslant p<1,\ f\in\mathcal{F}.\end{cases}\qquad(3)$$

В качестве семейства $\widetilde{A}_{(d)}$, $d\in\mathcal{U}$, выбрано семейство «определенных четких» (ОЧ) множеств, эквивалентных обычным множествам $A_{(d)}$, $d\in\mathcal{U}$. Их индикаторные функции имеют вид

$$\tau_{Uf}^{\widetilde{A}_{(d)}}(p)=\begin{cases}1, & \text{если } \begin{cases}p=1, & d\neq Uf,\\ p=0, & d=Uf,\end{cases}\\ 0, & \text{если } 0<p<1.\end{cases}\quad d,\,Uf\in\mathcal{U}.\qquad(4)$$

Оптимальное правило интерпретации $d^*(\cdot):\mathcal{R}_n\to\mathcal{U}$ определяется как решение двукритериальной задачи

$$\begin{aligned}&\sup_{a\geqslant n}Pl(N(d(\cdot))=a)\sim\min_{d(\cdot):\mathcal{R}_n\to\mathcal{U}},\ n\in[0,\,1],\\ &\sup_{a\leqslant n}Pl(N(d(\cdot))=a)\sim\min_{d(\cdot):\mathcal{R}_n\to\mathcal{U}},\ n\in[0,\,1],\end{aligned}\qquad(5)$$

если при решении задачи НН интерпретации необходимость — более важная характеристика качества интерпретации, чем правдоподобие. Оценка $d^*(\cdot)$ минимизирует правдоподобие больших и

максимизирует правдоподобие малых необходимостей ошибки оценивания НН элемента $U\widetilde{\varphi}$ НН элементом $d(\widetilde{\xi})$.

Оптимальное правило интерпретации $d^+(\cdot) : \mathcal{R}_n \to \mathcal{U}$ определяется как решение двукритериальной задачи

$$\begin{array}{l} \sup_{a \geqslant n} Pl(N(d(\cdot)) = a) \sim \min_{d(\cdot):\mathcal{R}_n \to \mathcal{U}},\ n \in [0,\ 1], \\ \inf_{a \geqslant n} Pl(N(d(\cdot)) = a) \sim \min_{d(\cdot):\mathcal{R}_n \to \mathcal{U}},\ n \in [0,\ 1], \end{array} \tag{6}$$

если правдоподобие — более важная характеристика качества интерпретации, чем необходимость, и, наконец, оптимальное правило $d_+(\cdot) : \mathcal{R}_n \to \mathcal{U}$ — решение двукритериальной задачи

$$\begin{array}{l} \sup_{a \geqslant n} Pl(N(d(\cdot)) = a) \sim \min_{d(\cdot):\mathcal{R}_n \to \mathcal{U}},\ n \in [0,\ 1], \\ \inf_{a \leqslant n} Pl(N(d(\cdot)) = a) \sim \min_{d(\cdot):\mathcal{R}_n \to \mathcal{U}},\ n \in [0,\ 1], \end{array} \tag{7}$$

если необходимость и правдоподобие — одинаково важные характеристики качества НН интерпретации.

Разрешимость задач (5), (6), (7) исследована в работе [3].

Как показано в [2], адекватность НН модели можно проверить, лишь если она удовлетворяет условиям:

$$\sup_{0 \leqslant p \leqslant 1} \tau_x^{\widetilde{\xi}}(p) = 1; \tag{8}$$

$$\left.\begin{array}{l} \text{если } \sup\limits_{0 < p \leqslant 1} \tau_x^{\widetilde{\xi}}(p) = 1,\ \text{то } \tau_x^{\widetilde{\xi}}(0) < 1; \\ \text{если } \tau_x^{\widetilde{\xi}}(0) = 1,\ \text{то } \sup\limits_{0 < p \leqslant 1} \tau_x^{\widetilde{\xi}}(p) < 1. \end{array}\right\} \tag{9}$$

Согласно условию (8) модель НН элемента $\widetilde{\xi}$ должна быть в достаточной степени определенной, т.е. для каждого значения $x \in \mathcal{R}_n$ НН элемента должно существовать вполне правдоподобное значение $p \in [0,\ 1]$. Условие (9) исключает противоречивые модели, для которых вполне правдоподобны возможные и одновременно невозможные значения $\widetilde{\xi}$.

Численный эксперимент На рис. 2 приведены результаты численного моделирования решения задачи интерпретации измерения, проведенного на основе разработанного на базе платформы Matlab программного комплекса. На рис. 2 а) изображен сигнал f, поданный на вход измерительного прибора, матрица аппаратной функции которого изображена на рис. 2 б); на рис. 2 в) изображен результат измерения $\widetilde{\xi} = x$, полученный по схеме (1). В НН моделях погрешности измерения $\widetilde{\nu}$ и входного сигнала $\widetilde{\varphi}$ $\bar{u}_1 = \bar{u}_2 = \bar{v}_1 = \bar{v}_2 = 20$, $\underline{u}_1 = \underline{u}_2 = \underline{v}_1 = \underline{v}_2 = 10$, $F = \beta^2 I$, $\beta^2 = 10$, I — единичная матрица размера 100×100. На рис. 2 г) и рис. 2 ж) сплошная линия — оценка входного сигнала $\widehat{f}$, полученная как решение задачи НН интерпретации (5), совпадающее с решением задач НН интерпретации (6), (7), пунктирная линия — входной сигнал f, при $\Sigma = \sigma^2 I$, $\sigma^2 = 10$ и $\sigma^2 = 0.1$ соответственно.

На рис. 2 д),е),з),и) изображены диаграммы, иллюстрирующие решение задачи проверки адекватности модели. На каждом рисунке изображены области $\widehat{X}$, $\widehat{\widehat{X}}$, $\check{X}$, и точка x_0 с координатами $(\|\Sigma^{-1/2}(x - A\widehat{f})\|^2, \|F^{-1/2}(\widehat{f} - f_0)\|^2)$, имеющими разные значения на каждом из рис. 2 д),е),з),и); значение $\|\Sigma^{-1/2}(x - A\widehat{f})\|^2$ отложено по горизонтальной, значение $\|F^{-1/2}(\widehat{f} - f_0)\|^2$ — по вертикальной оси. Взаимное расположение точки x_0 и областей $\widehat{X}$, $\widehat{\widehat{X}}$, $\check{X}$ позволяет судить об адекватности модели измерения. На рис. 2 д) $x_0 \in \widehat{X}$, при этом $x_0 \notin \widehat{\widehat{X}}$, поэтому результат измерения $\widetilde{\xi}$ не противоречит модели, для которой получена оценка $\widehat{f}$ на рис. 2 г). На рис. 2 е) модель не допускает проверки на адекватность, поскольку $x_0 \notin \widehat{X} \cup \check{X}$ (для такой модели результат интерпретации не приведен). На рис. 2 з) модель сомнительна, т.к. $x_0 \in \widehat{\widehat{X}}$ (для этой модели результат интерпретации приведен на рис. 2 ж)), причем чем ближе точка x_0 находится к внешней границе области $\widehat{\widehat{X}}$, тем более сомнительна модель, т.к. равная нулю возможность одновременного выполнения равенств $\widetilde{\xi} = x$ И $\widetilde{\varphi} = \widehat{f}$ более правдоподобна. Наконец, на рис. 2 и) $x_0 \in \check{X}$, поэтому результат наблюдения $\widetilde{\xi}$ противоречит модели измерения, и ее следует признать неадекватной (результат интерпретации не приведен).

Данных, представленных на рис. 2 достаточно, чтобы в диалоге с компьютером получить ответы на любые вопросы о качестве интерпретации и об адекватности модели измерения. Если при

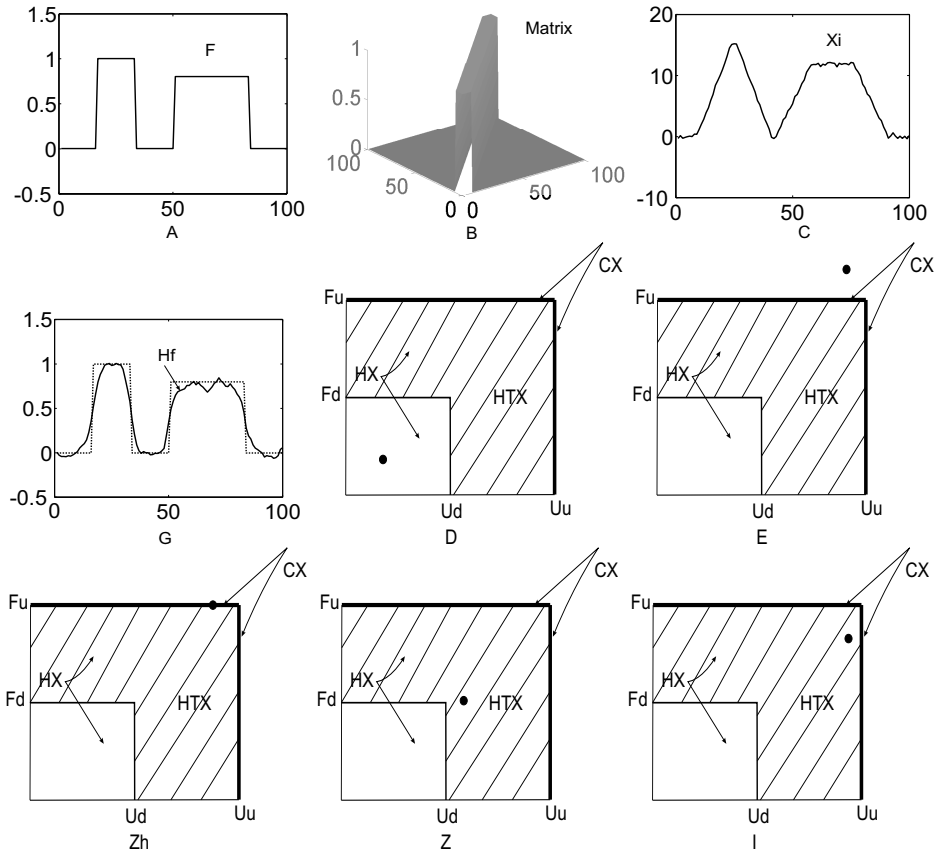


Рис. 2:

решении задачи интерпретации исследователь, получив решение, сталкивается с ситуациями, изображенными на рис. 2. е),з),и), он понимает, что нужно изменить параметры модели таким образом, чтобы модель стала непротиворечивой, и программный комплекс предоставляет ему такую возможность.

Обозначения

$$(\tau_1 \vee \tau_2)(p) \triangleq \sup\{\min(\tau_1(a), \tau_2(b)) \,|\, a,\, b \in [0,1],\ \max(a,\, b) = p\},$$
$$(\tau_1 \wedge \tau_2)(p) \triangleq \sup\{\min(\tau_1(a), \tau_2(b)) \,|\, a,\, b \in [0,1],\ \min(a,\, b) = p\}, \quad p \in [0,\, 1].$$

Список литературы

1. Пытьев Ю. П. Стохастические и нечеткие модели. Эмпирическое построение и интерпретация. Сборник трудов 1-й международной научно-практической конференции «Современные информационные технологии и ИТ-образование», 2005, с. 482–492.

2. Пытьев Ю. П. Неопределенные нечеткие модели и их применения. *Интеллектуальные системы* **8** (2004), вып. 1-4, 147–310.

3. Фаломкина О. В. Исследование нечетких и неопределенных нечетких методов анализа и интерпретации данных. Дисс. раб. на соиск. степ. канд. ф.-м. наук.

Параметрическая отказоустойчивость и оптимизация в графовых моделях дискретных систем [1]

Салий В. Н.,
e-mail: SaliiVN@info.sgu.ru

Пусть $G = (V, \alpha)$ - ориентированный граф (далее: граф) с множеством вершин V и отношением смежности α (см., например, [1]). Пусть, далее, $\Pi = (\pi_1, \pi_2, \ldots, \pi_k)$ - некоторый набор графовых параметров. Через $\Pi_0 = (\pi_1^0, \pi_2^0, \ldots, \pi_k^0)$ обозначим одну из допустимых конкретизаций набора Π, а через $\Pi(G)$ - его конкретизацию, соответствующую графу G. Граф G называется Π_0-графом, если $\Pi(G) = \Pi_0$. Естественно возникают следующие задачи.

I. *Для заданного* Π_0 *описать строение* Π_0*-графов.*

Множество дуг $\beta \subseteq \alpha$ назовем редуцируемым в Π_0-графе G, если $\Pi(G - \beta) = \Pi_0$, и нередуцируемым в противном случае. Π_0-граф, не имеющий редуцируемых множеств дуг, по определению является неприводимым. Заметим, что в случае неориентированных графов, т.е. когда отношение α симметрично и антирефлексивно, β также должно быть симметричным (удаляются ребра - пары встречных дуг).

Граф $G' = (V', \alpha')$ называется частью графа $G = (V, \alpha)$, если $V' \subseteq V$ и $\alpha' \subseteq \alpha$.

II. *Найти все (или некоторые, подчиняющиеся дополнительным условиям) неприводимые части заданного* Π_0*-графа* G*.*

Каждую такую часть можно назвать ядром Π_0-графа G. Дополнительными условиями могут быть, например, связность, минимальное возможное число остающихся дуг (ребер) и т.п.

Интерпретируя отказ в системе как удаление дуги в графе, моделирующем эту систему, видим, что редуцируемые множества дуг - это допустимые с точки зрения качеств Π_0 множества отказов. В этом контексте Π_0-ядра системы воспринимаются как наиболее важные ее подсистемы. Безопасность входящих в них системных связей требует особого внимания.

III. *Для заданных* G *и* Π_0 *найти (если это возможно) такие части* G' *графа* G*, чтобы* $\Pi(G') = \Pi_0$.

Каждую такую часть можно называть Π_0-редуктом графа G. Дополнительными условиями для G' могут быть остовность, связность, максимальное возможное число остающихся дуг (ребер) и т.п. Если $\Pi_0 = \Pi(G)$, то очевидным примером Π_0-редукта будет всякое ядро графа G.

IV. *Для заданных* G *и* Π_0 *найти (если это возможно) граф* $G' = (V, \alpha')$*, где* $\alpha \subseteq \alpha'$*, такой, чтобы* $\Pi(G') = \Pi_0$.

Каждый такой граф G' можно назвать Π_0-расширением графа G. Наиболее интересным представляется случай, когда G' имеет минимальное возможное число добавочных дуг (ребер).

Выделение максимальных редуктов или минимальных расширений графа, возникающих в задачах III и IV, можно рассматривать как оптимальные реконструкции моделируемой им системы, имеющие целью добиться желаемых качеств Π_0.

Из числа параметров, связываемых с графом, выделим его индекс и период, определяемые следующим образом. Пусть A - матрица смежности графа G. Наблюдая последовательность различных ее степеней $A, A^2, A^3, \ldots$, заметим, что эта последовательность конечна и что, если A^m - ее последний элемент, то $A^{m+1} = A^l$ для некоторого $l \leq m$. Число $\mathrm{ind}(A) = l - 1$ называется индексом матрицы A, а число $\mathrm{p}(A) = (m + 1) - l$ - ее периодом. По определению, $\mathrm{ind}(G) = \mathrm{ind}(A)$ и $\mathrm{p}(G) = \mathrm{p}(A)$, - индекс и период графа G.

Граф G называется идемпотентным, если $\mathrm{ind}(G) = 0, \mathrm{p}(G) = 1$, т.е. если идемпотентна его матрица смежности.

В настоящем сообщении задачи I-IV рассматриваются для случая $\Pi = (\mathrm{c}, \mathrm{ind}, \mathrm{p})$, где c - число (компонент) связности графа, и $\Pi_0 = (1, 0, 1)$, т.е. $\Pi(G) = \Pi_0$ означает, что G - связный идемпотентный граф. Один из подходов к решению задачи I для $\Pi = (\mathrm{ind}, \mathrm{p})$ и $\Pi_0 = (0, 1)$ предлагается в [3]. В [2] описаны (ind, p)-ядра для ациклических и функциональных графов.

Список литературы

[1]. Богомолов А.М., Салий В.Н. Алгебраические основы теории дискретных систем.- М.: Наука, 1997.

[2]. Салий В.Н. Отказоустойчивость и оптимизация дискретных систем с заданными индексом и периодом //Вестник Томского государственного университета.- Томск: ТГУ, 2006 (в печати).

[3]. Chaudhuri R., Mukherjea A. Idempotent Boolean matrices //Semigroup Forum.- 1980.- v. 21.- P. 273-282.

[1] Работа поддержана грантом РФФИ 05-08-18082.

Об отношении границ на конечных автоматах

Самоненко И. Ю., *e-mail: samonenko@yandex.ru*
кафедра Математической Теории Интеллектуальных Систем,
механико-математический факультет МГУ им. М. В. Ломоносова.

В данной работе рассматривается отношение границ на состояниях конечного автомата, сохраняемое при переходах автомата. Пусть **A** = (A,Q,φ) - конечный автомат (без выхода), где A - конечный входной алфавит, Q - конечное множество состояний и φ: Q × A → Q - функция переходов. Пусть $R \subseteq Q^{r+1}$ - некоторое отношение на множестве Q порядка r+1, обозначим $[q_1,\ldots,q_r]_R = \{q \in Q | (q,q_1,\ldots,q_r) \in R\}$.

Определение. Отношение $R \subseteq Q^{r+1}$ называется *отношением границ порядка r* на автомате **A** = (A,Q,φ) если:

1. Для любой перестановки $\sigma \in S_r$ верно $q \in [q_1,\ldots,q_r]_R \Rightarrow q \in [q_{\sigma(1)},\ldots,q_{\sigma(r)}]_R$
2. Для любых $q,q' \in Q$ верно $q \in [q',\ldots,q']_R \Rightarrow q = q'$
3. Существуют $g_1,\ldots,g_r \in Q$, такие что для любого $q \in Q$ верно $q \in [g_1,\ldots,g_r]_R$
4. Для любого $x \in A$ верно $q \in [q_1,\ldots,q_r]_R \Rightarrow \varphi(q,x) \in [\varphi(q_1,x),\ldots,\varphi(q_r,x)]_R$

Автомат, на котором можно ввести отношение границ порядка r, называется граничным *автоматом порядка r*. Через Γ_n^r обозначим класс всех граничных автоматов порядка r с n состояниями. Через K_n обозначим класс всех автоматов с n состояниями.

Теорема 1. $\Gamma_n^2 \subset \Gamma_n^3 \subset \ldots \subset \Gamma_n^{n-1} \subset \Gamma_n^n = K_n$. *При этом все включения строгие.*

Следствие 1. *Любой автомат является граничным для подходящего r.*

Теорема 2. *Существует алгоритм, который проверяет, является ли автомат граничным порядка r. Сложность алгоритма не превышает* n^{r+1}.

Автомат **A** = (A,Q,φ) называется синхронизуемым, если существуют слово $\alpha \in A^*$ и состояние $q_f \in Q$, такие, что для любого $q \in Q$, справедливо $\varphi(q,\alpha) = q_f$. В этом случае, слово α называется синхронизующим.

Теорема 3. *Пусть автомат* $\boldsymbol{A} \in \Gamma_n^r$ *является синхронизуемым, тогда минимальное по длине синхронизующее слово имеет длину* $\le (r-1)n(n-1)/2$.

Теорема 3 связана с гипотезой Черни, которая утверждает, что минимальное (по длине) синхронизующее слово имеет длину $\le (n-1)^2$ [2]. Это проблема до сих пор остается открытой. Наилучшие оценки сверху длины минимального синхронизующего слова имеют порядок $O(|Q|^3)$ [3].

Следствие 2. *Оценка длины минимального синхронизующего слова для класса граничных автоматов порядка 2 улучшает оценку Черни, для класса граничных автоматов порядка 3 асимптотически с ней совпадает, для класса граничных автоматов порядка* $r \ge 4$ *совпадает с ней по порядку.*

Рассмотрим сложность поиска синхронизующего слова и минимального синхронизующего слова для класса граничных автоматов.

Теорема 4. *Пусть автомат* $A \in \Gamma_n^r$ *является синхронизуемым, тогда существует алгоритм поиска синхронизующего слова, сложность которого не превышает* $(r-1)n(n-1)/2$.

Известно, что в общем случае, задача поиска минимального синхронизующего слова является NP-полной [1]. Для класса граничных автоматов, при фиксированном порядке r, эта задача становится полиномиальной.

Теорема 5. *Пусть автомат* $A \in \Gamma_n^r$ *является синхронизуемым, тогда существует алгоритм поиска минимального синхронизующего слова, сложностью* $O(n^r)$.

В работе рассмотрена система структурных операций над граничными автоматами, которые сохраняют свойств автоматов быть граничными. Приведены примеры классов граничных автоматов (монотонные, линейные, групповые, и д.р.) Так же вводится понятие *отношения влияний* на конечном автомате, которое является обобщением отношения границ, и позволяет эффективно решать различные задачи связанные с конечными автоматами.

Автор выражает глубокую благодарность своему научному руководителю профессору Бабину Д. Н. за постановку задачи и помощь в ее решении.

Список Литературы

[1] David Eppstein "Reset Sequences for Monotonic Automata", 1990.

[2] Cerny, "Poznamka k homogenum eksperimentom s konechnymi automatami", Math.-Fiz. Cas., 14(1964)

[3] A. N. Trahtman "The existence of synchronizing word and Cerny conjecture for some finite automata"

Анализ графов с помеченными вершинами

Сапунов С. В.,
Институт прикладной математики и механики НАН Украины, Донецк

Задачи организации двигательного поведения или навигации автономных мобильных роботов являются одними из основных задач искусственного интеллекта [1]. Одной из центральных проблем навигации является "проблема самолокализации". Эта проблема состоит в следующем: задан конечный граф (карта). Требуется найти такое множество путей по графу, которое позволило бы роботу, установленному в произвольную вершину, определить ее.

Конечным, простым, неориентированным, инициально связным графом с помеченными вершинами назовем пятерку $\mathcal{G} = (G, E(G), M, \mu, g_0)$, где G — множество вершин, $E(G) \subseteq ((g,h) \,|\, g,h \in G \wedge g \neq h)$ — множество ребер, M — множество меток, $\mu : G \to M$ — сюръективная функция разметки. Последовательность меток вершин $w = \mu(g_1)\dots\mu(g_k)$, соответствующую некоторому пути $g_1 \dots g_k$ в графе $\mathcal{G}$, назовем словом. Через w^{rev} обозначим слово $\mu(g_k)\dots\mu(g_1)$. Определим язык L_g как множество всех слов, порождаемых вершиной $g \in G$. Через $L_{\mathcal{G}}$ обозначим множество языков вершин $\bigcup\limits_{g\in G} L_g$ и назовем его языком, порожденным графом $\mathcal{G}$. Введем операцию $\star : G \times M^+ \to G$ соотношением: для любой вершины $g \in G$ и любого слова $w \in M^+$ через $g \star w$ обозначим вершину $h \in G$ такую, что существует путь, соединяющий вершины g и h, и иетка этого пути равна w. Для слов $u, w \in M^+$ введем их композицию $u \circ w$ равную uw' если $u = u'x$, $w = xw'$, $x \in M$, и не определенную в противном случае.

Будем говорить, что вершины $g, h \in G$ неотличимы и писать $(g,h) \in \varepsilon$ если $L_g = L_h$.

Открытой окрестностью $O_{(g)}$ вершины $g \in G$ называется множество всех вершин графа $\mathcal{G}$ такое, что $O_{(g)} = \{h \,|\, \{g,h\} \in E(G)\}$. Закрытой окрестностью O_g, соответствующей $O_{(g)}$, называется объединение $O_{(g)} \cup \{g\}$. Граф $\mathcal{G}$ назовем сильно детерминированным графом или SD-графом, если для любой вершины $g \in G$ и любых вершин $s, t \in O_{[g]}$ из $s \neq t$ следует, что $\mu(s) \neq \mu(t)$. Из определения вытекает следующее утверждение.

Теорема 1. *Если g — произвольная вершина SD-графа $\mathcal{G}$, то $L_g \subseteq L_{G-\{g\}}$ тогда и только тогда, когда существует вершина $h \in G - \{g\}$ такая, что $L_g \subseteq L_h$.*

Граф G назовем приведенным если $L_g \neq L_h$ для всех $g, h \in G$, $g \neq h$.

Будем говорить, что вершина $h \in G$ покрывает вершину $g \in G$ и писать $(g,h) \in \varkappa$, если $L_g \subseteq L_h$. Отношение $\varkappa$ рефлексивно, транзитивно, но в общем случае не антисимметрично и, таким образом, является предпорядком. Ясно, что $\varkappa \cap \varkappa^{-1} = \varepsilon$. Вершину g назовем максимальной, по $\varkappa$ в графе $\mathcal{G}$, если для всех $h \in H$ из $(g,h) \in \varkappa$ следует, что $(g,h) \in \varepsilon$. Множество всех максимальных вершин обозначим через $G^{\max}$. В [2] показано, что множество всех максимальных вершин SD-графа $\mathcal{G}$ образует подграф $\mathcal{G}^{\max} \subseteq \mathcal{G}$ и $L_{\mathcal{G}} = L_{\mathcal{G}^{\max}}$. Там же показано, что максимальный подграф $\mathcal{G}^{\max}$ приведенного SD-графа $\mathcal{G}$ является наименьшим по числу вершин графом в классе всех графов с одним и тем же языком $L_{\mathcal{G}}$. Далее, если не оговорено противное, рассматриваются только приведенные, максимальные SD-графы.

Конечное множество слов $I_g \subseteq L_g$ назовем идентификатором локализации вершины g или, короче, идентификатором вершины g, если для любой вершины $h \neq g$, $h \in G$, выполняется $I_g - L_h \neq \varnothing$. Число слов в множестве I_g назовем кратностью идентификатора. Идентификатор назовем простым, если его кратность равна 1. Наибольшую из длин слов в I_g назовем высотой идентификатора.

Из теоремы 1 и определения приведенного максимального SD-графа следует, что для любой вершины $h \in G$, $h \neq g$, существует слово $w_h \in L_g - L_h$. Тогда множество слов $\bigcup\limits_{\substack{h\in G\\ h\neq g}} w_h$ является кратным идентификатором вершины g. Обозначим через $\mathfrak{I}_g$ класс всех идентификаторов I_g вершины g. Очевидно, что для любой неизолированной вершины g приведенного максимального SD-графа $\mathcal{G}$ класс $\mathfrak{I}_g$ бесконечен. Действительно, всякому слово w, принадлежащее некоторому идентификатору I_g, в силу симметричности $\mathcal{G}$ можно заменить словом $w \circ w^{rev}$ и получить, таким образом, новый идентификатор вершины g.

Пусть $L_1 \prec L_2$ означает, что каждое слово из L_1 является начальным отрезком некотрого слова из L_2. Это отношение является предпорядком и порождает эквивалнтность $\equiv$. Через $\mathfrak{K}(I_g)$ обозначим класс всех идентификторов из $\mathfrak{I}_g$ эквивалентных (по $\equiv$) идентификатору I_g. Идентификатор I_g назовем минимальным в $\mathfrak{I}_g$, если для всех $I'_g \in \mathfrak{I}_g$ из $I'_g \prec I_g$ следует $I_g \subseteq I'_g$.

Теорема 2. *Равносильны утверждения:*
1. I_g минимален в $(\mathfrak{I}_g, \prec)$;

2. I_g минимален в $(\mathfrak{K}(I_g), \subseteq)$ и I_g минимален в $\langle \mathfrak{I}_g/_{\equiv}, \prec \rangle$;

3. множество I'_g, полученное из I_g удалением хоть одного слова или заменой хоть одного слова его собственным начальным отрезком, идентификатором вершины g не является.

Следующее утверждение оценивает высоту минимальных идентификаторов.

Теорема 3. *Для любых $n \geqslant 6$ и $m \geqslant 4$ существует приведенный, максимальный SD-граф с n вершинами и m метками и его вершина g, для которой найдется минимальный идентификатор как угодно большой высоты.*

Эта теорема показывает, что множество минимальных идентификаторов $\mathfrak{K}^{\min}(I_g) \subseteq \mathfrak{K}(I_g)$ в общем случае бесконечно.

Рассмотрим интересное свойство идентификаторов вершин. Пусть $I_g = \{w_1, \ldots, w_l\}$ — некоторый идентификатор вершины g SD-графа $\mathcal{G}$. По каждому слову $w_i = x_{i_1} \ldots x_{i_k}$, $1 \leq i \leq l$ построим ациклический помеченный неорграф $\mathcal{T}(w_i)$, множество вершин которого равно $\{t_{i_1}, \ldots, t_{i_k}\}$, множество ребер состотит из всех пар $(t_{i_j}, t_{i_{j+1}})$, $1 \leq j < k$, а метка $\mu(t_{i_j})$ равна x_{i_j}. Вершину t_{i_1} назовем начальной. В графе $\mathcal{T}(I_g) = \sum_{i=1}^{l} \mathcal{T}(w_i)$ отождествим все начальные вершины в вершину t_1 и детерминизируем полученный граф, то есть многократно применим следующую операцию: если в окрестность некоторой вершины попадают одинаково помеченные вершины, то такие вершины отождествим, заменяя возникающие кратные ребра одним ребром. Очевидно, что корневое дерево $\mathcal{T}(I_g)$ является SD-графом и $I_g \subseteq L_{\mathcal{T}(I_g)}$.

Множество слов $Q = \{u_1, \ldots, u_k\}$ назовем обходом графа $\mathcal{T}(I_g)$ если 1) любое слово $u_i \in Q$ есть метка некотрого пути из начальной вершины по графу $\mathcal{T}(I_g)$; 2) для любой вершины t графа $\mathcal{T}(I_g)$ существует слово $u \in Q$, такое, что его начальный отрезок u' соответствует пути из начальной вершины t_1 в вершину t.

Теорема 4. *Любой обход Q корневого дерева $\mathcal{T}(I_g)$ является идентификатором вершины g.*

Выберем проивольную вершину t графа $\mathcal{T}(I_g)$. Пусть p — путь из корня в t и слово w является его меткой. Справделиво следующее утверждение.

Теорема 5. *Любой обход Q корневого дерева $\mathcal{T}(I_g)$ с корнем в вершине t является идентификатором вершины $g \star w$ в графе $\mathcal{G}$.*

В этом заключается существенное отличие идентификаторов вершин SD-графов от начальных идентификаторов состояний конечных автоматов [3], для которых подобное в общем случае не имеет места.

Список литературы

[1.] Borenstein J. Everett B., Feng L. Navigation Mobile Robots: System and Techniques. – A.K. Peters, Ltd., Wellesley, MA, – 1996, – 223 p.

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

[3.] Грунский И. С. Анализ поведения конечных автоматов. – Луганск: Изд-во ЛГПУ, 2003, – 318 с.

Критерии Бухбергера и тривиальные сизигии[1]

Семенов А. С.,

Московский государственный университет, механико-математический факультет

e-mail: semyonov1980@mail.ru

В настоящее время многими исследователями ведется работа по нахождению путей ускорения процесса вычисления базиса Гребнера по сравнению с уже существующими алгоритмами. На этом

[1]Работа была частично поддержана Российским Фондом Фундаментальных Исследований, проект №05-01-00671

пути был достигнут ряд успехов. Вместе с тем, возникает необходимость сравнения и систематизации этих решений, поскольку каждое из них содержит свою терминологию, что делает задачу сопоставления с другими алгоритмами весьма затруднительной.

Базовым алгоритмом конструктивного построения базиса Гребнера является алгоритм Бухбергера. Далее он будет сформулирован в традиционном виде, в каком он приведен в [1]. Для оптимизации используются два критерия Бухбергера, позволяющие исключать из рассмотрения ряд S-полиномов. В алгоритме полиномы в G нумеруются, и пары различных полиномов f_i, f_j обозначены $(i,j), i < j$. В списке B находятся те пары, для которых соответствующие им S-полиномы должны быть вычислены и отредуцированы в ходе алгоритма.

Определение 1 *Путь $\prec$ — допустимое отношение порядка на моноиде $\mathbb{M}$. Наибольший относительно $\prec$ моном $u \in \mathbb{M}$, входящий в полином f, называется старшим мономом f и обозначается* $\mathrm{lm}(f)$. *Коэффициент при* $\mathrm{lm}(f)$ *называется старшим коэффициентом f и обозначается как* $\mathrm{lc}(f)$. *Старший член* $\mathrm{lc}(f)\,\mathrm{lm}(f)$ *обозначается, как* $\mathrm{lt}(f)$. *По аналогии, можно определить второй по старшинству член в f, который обозначается как* $\mathrm{sc}(f)\,\mathrm{sm}(f)$.

Моном $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$ обозначается как $\mathrm{lcm}(f,g)$, где lcm — наименьшее общее кратное. Наибольший общий делитель $\mathrm{lm}(f), \mathrm{lm}(g)$ обозначается как $\gcd(f,g)$.

Первый критерий Бухбергера состоит в том, что если $\mathrm{lcm}(f,g) = \mathrm{lm}(f)\,\mathrm{lm}(g)$, то такая пара может быть исключена из рассмотрения в ходе алгоритма.

Второй критерий Бухбергера, в свою очередь, выполнен, если истинна функция $Criterion(f_i, f_j, B)$. Для ее определения вводится новое обозначение для пар

$$[i,j] = \left[\begin{array}{ll} (i,j), & i < j, \\ (j,i), & j < i. \end{array} \right.$$

Логическая функция $Criterion(f_i, f_j, B)$ истинна, если $\exists l \notin \{i,j\}$, для которого $[i,l], [j,l]$ не принадлежат B и $\mathrm{lm}(f_l) | \mathrm{lcm}(f_i, f_j)$.

Алгоритм Бухбергера
вход: конечное множество полиномов F
выход: базис Гребнера G идеала $I = Id(F)$

```
(f_1, ..., f_s) := Авторедукция(F)
G := (f_1, ..., f_s)
B := {(i,j) | 1 ≤ i < j ≤ s}
t := s
пока B ≠ ∅
  выбрать (i,j) ∈ B
  если lcm(f_i, f_j) ≠ lm(f_i) lm(f_j) и ¬Criterion(f_i, f_j, B) то
    S := NF(S(f_i, f_j), G)
    если S ≠ 0 то
      t := t + 1
      f_t := S
      G = G ∪ {f_t}
      B := B ∪ {(i,t) | 1 ≤ i ≤ t − 1}
  B := B − {(i,j)}
конец
вернуть G
конец
```

В алгоритме остается неопределенным правило выбора текущей пары $(i,j) \in B$. Правило, согласно которому выбирается пара с наименьшим возможным $\mathrm{lcm}(f_i, f_j)$ относительно некоторого допустимого порядка $\sqsubset$ (не обязательно совпадающего с мономиальным упорядочением $<$), называется *нормальной стратегией*.

В процессе выполнения данного алгоритма возникает задача — отбросить как можно большее количество S-пар, про которые заранее известно, что они впоследствии отредуцируются к 0.

Существует два подхода к исключению нулевых редукций — использование отношений сравнимости и использование сизигий.

Определение 2 *Если полином f может быть представлен в виде суммы $h_1 g_1 + \ldots + h_s g_s$, где h_i — полиномы, g_i полиномы из множества G, и выполнено* $\mathrm{lm}(f) \succeq \mathrm{lm}(h_i g_i)$, *то*

$$f \equiv 0 \pmod{G}$$

Если $f - g \equiv 0 \pmod G$, то $f \equiv g \pmod G$. Отношение $\equiv$ является рефлексивным, симметричным и транзитивным, следовательно, является отношением эквивалентности.

Заметим, что если полином редуцируется к 0 по G, то он сравним с 0 по G, но представление полинома f в виде суммы $h_1g_1 + \ldots + h_sg_s$ ($\mathrm{lm}(f) \succeq \mathrm{lm}(h_ig_i)$) автоматически не влечет за собой редукции f к 0 по G. Впрочем, если G — базис Гребнера порожденного им идеала G, то $f \equiv 0 \pmod G$ влечет за собой соотношение $NF(f, G) = 0$. На этом свойстве и основывается обоснование корректности критериев Бухбергера.

В работах [2,3] показано, что количество критических пар, которые можно не использовать в ходе алгоритма Бухбергера, может существенно превышать количество тех, что отбрасываются благодаря использованию критериев. Для этих целей авторами используется специальная терминология, использующая полиномиальные сизигии.

Определение 3 *Пусть g_i — полиномы из множества G ($i = 1, \ldots, n$). Тогда совокупность n-полиномов $(h_1, \ldots, h_n)$ называется сизигией, если имеет место равенство $h_1g_1 + \ldots + h_ng_n = 0$.*

Очевидно, что если S-полином $S(g_i, g_j)$ был отредуцирован к 0 по G, то процесс формирования S-полинома и последующая его редукция к 0, записанные как последовательность сложений и вычитаний полиномов, соответствуют некоторой сизигии.

В [2,3] показано, что ряд критических пар можно не рассматривать за счет использования "тривиальных" сизигий $g_jg_i - g_ig_j$, где g_i, g_j — различные полиномы из G. Следовательно возникает задача расширить и усилить критерии Бухбергера, введя в них "информацию" содержащуюся в вышеприведенных "тривиальных" равенствах, и понять насколько значительный эффект даст новая информация. Данная задача может решена при помощи полиномиальных сравнений лишь частично.

Лемма 1 *Пусть G — полиномиальное множество, и g_1, g_2 — два различных полинома, для которых имеет место $\mathrm{sm}(g_1)\,\mathrm{lm}(g_2) \neq \mathrm{sm}(g_2)\,\mathrm{lm}(g_1)$. Тогда имеет место соотношение.*

$$\gcd(g_1, g_2)S(g_1, g_2) \equiv 0 \pmod G$$

Доказательство. Рассмотрим тождество $g_1g_2 - g_2g_1 = 0$. Пусть $g_1 = \mathrm{lc}(g_1)\,\mathrm{lm}(g_1) + p$, $g_2 = \mathrm{lc}(g_2)\,\mathrm{lm}(g_2) + q$. Очевидным следствием этого равенства является выражение

$$\mathrm{lc}(g_1)\,\mathrm{lm}(g_1)g_2 - \mathrm{lc}(g_2)\,\mathrm{lm}(g_2)g_1 = -p * g_2 + q * g_1.$$

Докажем, что

$$\mathrm{lm}(\mathrm{lc}(g_1)\,\mathrm{lm}(g_1)g_2 - \mathrm{lc}(g_2)\,\mathrm{lm}(g_2)g_1) = \max\{\mathrm{lm}(p * g_2), \mathrm{lm}(q * g_1)\}.$$

Из того, что $\mathrm{sm}(g_1) = \mathrm{lm}(p)$, $\mathrm{sm}(g_2) = \mathrm{lm}(q)$, и $\mathrm{sm}(g_1)\,\mathrm{lm}(g_2) \neq \mathrm{sm}(g_2)\,\mathrm{lm}(g_1)$ следует, что старшие члены $\mathrm{lm}(p * g_2)$ и $\mathrm{lm}(q * g_1)$ не сокращаются. Следовательно, выражение $-p * g_2 + q * g_1$ дает сравнимость с 0 по модулю G для левой части. Остается заметить, что $\mathrm{lc}(g_1)\,\mathrm{lm}(g_1)g_2 - \mathrm{lc}(g_2)\,\mathrm{lm}(g_2)g_1 = \gcd(g_1, g_2)S(g_1, g_2)$.

Лемма 2 *Если $\gcd(g_1, g_2) = 1$, то предыдущая лемма обеспечивает выполнение первого критерия Бухбергера, то есть $S(g_1, g_2) \equiv 0 \pmod G$.*

Доказательство. Для доказательства достаточно показать, что $\mathrm{sm}(g_1)\,\mathrm{lm}(g_2) \neq \mathrm{sm}(g_2)\,\mathrm{lm}(g_1)$. Действительно, в противном случае, поскольку $\gcd(g_1, g_2) = 1$, имеет место $\mathrm{lm}(g_1) | \mathrm{sm}(g_1)$, что невозможно, так как $\mathrm{lm}(g_1) \succ \mathrm{sm}(g_1)$.

Таким образом, первый критерий Бухбергера представляет собой ни что иное, как использование информации, содержащейся в тривиальных сизигиях. Тем не менее, переход от сизигий к более простому понятию сравнения в общем случае требует анализа дополнительной информации — вторых по старшинству членов в полиномах.

Замечание 1 *Лемму 1 можно усилить, поскольку она будет верна и в случае $\mathrm{sm}(g_1)\,\mathrm{lm}(g_2) = \mathrm{sm}(g_2)\,\mathrm{lm}(g_1)$ и $\mathrm{sc}(g_1)\,\mathrm{lc}(g_2) \neq \mathrm{sc}(g_2)\,\mathrm{lc}(g_1)$. Тем не менее, операции с коэффициентами часто занимают много времени и проверка последнего условия представляется затратной.*

Лемма 1 позволяет несколько расширить второй критерий Бухбергера и множество отбрасываемых пар.

Теорема 1 *Пусть G множество полиномов, и g_1, g_2, g_3 — три различных элемента из G, такие, что $\mathrm{lm}(g_3) | \mathrm{lcm}(g_1, g_2)$. Про каждую пару (g_i, g_3) ($i \in \{1, 2\}$, j — элемент, не равный i в $\{1, 2\}$) известно следующее:*

- *либо* $S(g_i, g_3) \equiv 0 \pmod{G}$

- *либо* $\operatorname{lm}(g_3) | \frac{\operatorname{lm}(g_i)}{\gcd(g_1, g_2)}$ *и* $\operatorname{sm}(g_j)\operatorname{lm}(g_3) \neq \operatorname{sm}(g_3)\operatorname{lm}(g_j)$.

Тогда $S(g_1, g_2) \equiv 0 \pmod{G}$.

Доказательство. Доказательство будет основыватья на тождестве

$$S(g_1, g_2) = \alpha \frac{\operatorname{lcm}(g_1, g_2)}{\operatorname{lcm}(g_1, g_3)} S(g_1, g_3) - \beta \frac{\operatorname{lcm}(g_1, g_2)}{\operatorname{lcm}(g_2, g_3)} S(g_2, g_3),$$

где α, β — коэффициенты.

Если оба S-полинома $S(g_1, g_3), S(g_2, g_3)$ сравнимы с 0, то доказательство очевидно. Остается рассмотреть выполнение второго условия.

Пусть $i = 2, j = 1$. Имеем равенства

$$\frac{\operatorname{lcm}(g_1, g_2)}{\operatorname{lcm}(g_1, g_3)} S(g_1, g_3) = \frac{\operatorname{lm}(g_1)\operatorname{lm}(g_2)/\gcd(g_1, g_2)}{\operatorname{lm}(g_1)\operatorname{lm}(g_3)/\gcd(g_1, g_3)} S(g_1, g_3) =$$

$$= \frac{\operatorname{lm}(g_2)}{\operatorname{lm}(g_3)\gcd(g_1, g_2)} \gcd(g_1, g_3) S(g_1, g_3).$$

В случае выполнения второго условия имеем

$$\frac{\operatorname{lcm}(g_1, g_2)}{\operatorname{lcm}(g_1, g_3)} S(g_1, g_3) \equiv 0 \pmod{G}.$$

Таким образом, второй критерий Бухбергера может быть расширен следующим образом.

Логическая функция $Criterion(f_i, f_j, B)$ истинна, если $\exists l \notin \{i, j\}$, для которого $\operatorname{lm}(f_l) | \operatorname{lcm}(f_i, f_j)$, где элемент $[i, l]$ не принадлежит B или же принадлежит B с выполнением условия

$$\operatorname{lm}(g_l) | \frac{\operatorname{lm}(g_i)}{\gcd(g_1, g_2)},$$

$$\operatorname{sm}(g_j)\operatorname{lm}(g_l) \neq \operatorname{sm}(g_l)\operatorname{lm}(g_j),$$

а для $[j, l]$ все аналогично при перемене местами индексов i, j между собой.

Благодаря условию $\operatorname{lm}(f_l) | \operatorname{lcm}(f_i, f_j)$ в случае **нормальной стратегии** и последовательного выполнения алгоритма Бухбергера расширение второго критерия не будут давать дополнительный выигрыш. Тем не менее, в случае параллелизации алгоритма Бухбергера и нарушения нормальной стратегии можно ожидать, что данный критерий выявит ряд новых "лишних" $S-$пар. В этой связи следует отметить, что стратегия алгоритма ([2]) не является нормальной, поскольку основана на последовательном вычислении базисов Гребнера идеалов $\langle f_n \rangle$, $\langle f_{n-1}, f_n \rangle$, $\ldots$, $\langle f_1, \ldots, f_n \rangle$.

Список литературы

1. Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. М. Мир, 2000.

2. Faugère J.-Ch. A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5). *Proceedings of ISSAC*,ACM Press (2002),75–83.

3. Möller H.M., Mora T., Traverso C. Gröbner Bases Computation Using Syzygies.*ISSAC 1992*, 320-328.

Вариация стайного управления группой объектов

Скобелев В. Г., *E-mail: skbv@iamm.ac.donetsk.ua*
Институт прикладной математики и механики НАН Украины, г. Донецк

На основе стайных принципов поведения исследуется задача моделирования сложных систем, состоящих из большого числа слабо взаимодействующих подсистем.

Ключевые слова: стайное управление, оптимизация поведения группы слабо взаимодействующих объектов, моделирование, автоматы в лабиринтах

Введение.

Исследование основ управления многоуровневыми иерархическими системами [1] привело к пониманию высокой сложности решения этих задач. Поэтому значительные усилия были направлены на разработку моделей и методов (часто плохо сравнимых друг с другом из-за различий в используемом математическом аппарате и постановках задач), предназначенных для реализации механизмов управления для узких классов систем (при наличии ограничений на их структуру). К ним относятся модели и методы управления распределенными системами на основе автоматных моделей и сетей Петри [2,3], попытки применения аппарата теории игр в экономике, политике и социальных науках [4-6]. Результат – определенные успехи при исследовании задач координации и интерактивного взаимодействия подсистем. Однако вне поля зрения исследователей остался широкий класс задач, общая характеристика которых – то, что предмет исследования – это группа (практически) не взаимодействующих друг с другом (практически) неразличимых объектов, связанных общей целью. В настоящее время задачи исследования таких систем постоянно возникают в военной области (разработка интеллектуального оружия, планирование мероприятий по предотвращению атак террористов и т.д.), обеспечение безопасности глобальных информационных систем (типа Internet), экономических систем и т.д. Многочисленные примеры таких задач можно найти в отчетах корпорации RAND, выпущенных в течение последнего десятилетия. Итак, разработка математических основ и техники компьютерного моделирования, предназначенных для исследования стратегий управления группой не взаимодействующих друг с другом неразличимых объектов, связанных общей целью актуальна в настоящее время. Важный подкласс указанных задач составляют те, в которых речь идет об исследовании принципов стайного управления. В этом случае отсутствие иерархии, интерактивного взаимодействия, практическая неразличимость объектов друг от друга и общая цель приводят к тому, что все многообразие стратегий управления, по своей сути, характеризуется многообразием режимов компьютерного моделирования действий стаи. Отсюда вытекает возможность разработки унифицированного механизма, предназначенного для управления скоростью адаптации к вариациям ситуации и оптимизирующего (по выбранному критерию) поведение стаи. Ясно, что исследование стратегий управления стаей – это основа для выделения основных принципов трансформации стаи в иерархическую (возможно, распределенную) систему. Подтверждение этого – исследование теоретических характеристик поведения групп автоматов в лабиринтах (играющих роль внешней среды) при различных предположениях на типы автоматов, уровень их взаимодействия и структуру лабиринта (см., напр., [7]).

В [8] стайное управление исследуется в рамках следующей математической модели. Стая – это множество объектов $O = \{o_1, \ldots, o_n\}$, функционирующих в среде E в дискретном времени. В момент t состояние среды характеризуется вектором $\mathbf{e}(t) = (e_1(t), \ldots, e_w(t))$, а состояние и действие объекта o_i $(i = 1, \ldots, n)$ – векторами $\mathbf{s}_i(t) = (s_1^{(i)}(t), \ldots, s_{m_i}^{(i)}(t))$ и $\mathbf{a}_i(t) = (a_1^{(i)}(t), \ldots, a_{d_i}^{(i)}(t))$. Координация состояний объектов с состоянием среды характеризуются неравенствами $\alpha_j(\mathbf{s}_1(t), \ldots, \mathbf{s}_n(t), \mathbf{e}(t)) \le 0$ $(j = 1, \ldots, k_1)$, а допустимые действия объектов в ситуации $(\mathbf{s}_1(t), \ldots, \mathbf{s}_n(t), \mathbf{e}(t))$ – неравенствами $\beta_j(\mathbf{s}_1(t), \ldots, \mathbf{s}_n(t), \mathbf{a}_1(t), \ldots, \mathbf{a}_n(t), \mathbf{e}(t)) \le 0$ $(j = 1, \ldots, k_1)$. В результате действия объекта может измениться состояние среды и состояния некоторых объектов стаи. Решается задача: $F(\mathbf{s}_1(t), \ldots, \mathbf{s}_n(t), \mathbf{a}_1(t), \ldots, \mathbf{a}_n(t), \mathbf{e}(t)) \to ext$ (F – заданный функционал) в некоторый момент t (возможно, удовлетворяющий заданным условиям). В [9] при условии, что состояние объекта определяеься только его координатами на плоскости исследованы задачи прикрытия кругового люка крышкой при равномерном распределении объектов (на крышке) и действиях, представленных композицией параллельного переноса и поворота (т.е. достижение неподвижной цели), а также уничтожения стаей круговой цели, совершающей периодические колебания. Показано, что минимизация времени и длины траектории – это эквивалентные задачи. Цель настоящей работы – исследование детерминированного и вероятностного подходов в процессе стайного управления.

Основные результаты.

По-видимому, в пространстве $\mathbf{R}^k$ общая модель в ситуации «стая-цель» – это представление каждого объекта стаи o_i $(i=1,\ldots,n)$ вектором $(x_1^{(i)}(t),\ldots,x_k^{(i)}(t),\alpha^{(i)}(t))$ ($(x_1^{(i)}(t),\ldots,x_k^{(i)}(t))$ – координаты, $\alpha^{(i)}(t)$ – потенциал o_i в момент t), а состояние цели – вектором $\mathbf{e}(t)=(e_1(t),\ldots,e_k(t),\beta(t))$ ($(e_1(t),\ldots,e_k(t))$ – координаты, $\beta(t)$ – потенциал цели в момент t). При любом действии объекта o_i $(i=1,\ldots,n)$, направленном на приближение к цели, его потенциал $\alpha^{(i)}(t)$ снижается на величину $\mu^{(i)}(t)$. При поражении цели объектом o_i его потенциал $\alpha^{(i)}(t)$ увеличивается на величину $\nu^{(i)}(t)$. Аналогичным образом, при любом действии цели, без ее поражения объектом стаи, ее потенциал снижается на величину $\kappa(t)$, а при поражении цели объектом o_i снижение потенциала цели осуществляется на величину $\kappa(t)+\lambda_i(t)$.

В результате компьютерного моделирования динамики действий стаи осуществляется переход от пространства $\mathbf{R}^k$ к пространству $(\mathbf{Q}_{a_1,a_2})^k$, т.е. к приближенному представлению действительных чисел рациональными числами (с фиксированной или плавающей запятой). Итак, для каждого $k\in\mathbf{N}$ в действиях, как каждого объекта стаи, так и в действиях цели имеется $2^{(a_1+a_2)^k}$ степеней свободы. В качестве действий объектов стаи и цели естественно ограничиться некоторым фиксированным классом алгоритмов A. Пусть A – класс о.-д. функций [10]. Имеет место следующая теорема

Теорема 1. *Для любых алгоритмов* $A_i\in\mathsf{A}$ $(i=1,\ldots,n)$, *определяющих действия объектов стаи* o_i $(i=1,\ldots,n)$ *и алгоритма* $B\in\mathsf{A}$, *определяющего действие цели, любая задача моделирования поведения стаи эквивалентна некоторой задаче поведения коллектива* n *автоматов в* $2^{(a_1+a_2)^k}$*-мерном нестационарном лабиринте.*

Для построения гибкой математической модели, предназначенной для решения широкого класса задач исследования поведения стаи $\bigcirc=\{o_1,\ldots,o_n\}$, следующие параметры должны быть внесены в компьютерную модель в явном виде:

1) параметры, обеспечивающие выбор режима моделирования действий объектов $\bigcirc=\{o_1,\ldots,o_n\}$;
2) параметры, обеспечивающие выбор порядка активизации объектов стаи;
3) параметры, обеспечивающие выбор предпочтений на множестве допустимых действий каждого объекта стаи $o_i\in\bigcirc$ $(i=1,\ldots,n)$.

Охарактеризуем эти группы параметров. Многообразие режимов моделирования поведения стаи может быть охарактеризовано мощностями $m(t)$ множеств объектов стаи, активируемых в момент t. Два экстремальных случая соответствуют специальным режимам моделирования:

1) если $m(t)=1$ в каждый момент t, то имеет место простой асинхронный режим моделирования поведения стаи;
2) если $m(t)=n$ в каждый момент t, то имеет место синхронный асинхронный режим моделирования поведения стаи.

Отсюда вытекает, что:

1) многообразие чисто асинхронных режимов моделирования поведения стаи определяется условием: $m(t)<n$ в каждый момент t;
2) многообразие смешанных асинхронно-синхронных режимов моделирования поведения стаи определяется условием: существует такой момент t_1, что $m(t_1)<n$ и существует такой момент t_2, что $m(t_2)=n$.

Пусть Δt – минимальный промежуток времени, достаточный для выполнения любого действия любым объектом $o_i\in\bigcirc$ $(i=1,\ldots,n)$, а T – максимальное время реализации одной итерации поведения стаи $\bigcirc$. Имеет место следующая теорема

Теорема 2. *1. Для простого асинхронного режима моделирования поведения стаи* $T\le\Delta t\cdot\max\limits_t\left|n\cdot m^{-1}(t)\right|$ *в каждый момент* t.

2. Для синхронного режима моделирования поведения стаи $T=\Delta t$ *в каждый момент* t.

3. Для любого смешанного асинхронно-синхронного режима моделирования поведения стаи $\Delta t \le T \le \Delta t \cdot \max_t \left| n \cdot m^{-1}(t) \right|$ *в каждый момент* t*, причем обе границы достижимы.*

В процессе моделирования поведения стаи упорядочение объектов может быть как фиксированным, так и гибким. В первом случае любой объект стаи активируется в заранее определенный момент. Во втором случае в качестве активируемого объекта выбирается тот, чье действие наиболее благоприятно для стаи, рассматриваемой как единое целое. Отсюда вытекает, что:

1) в явном виде может быть внесен механизм, оптимизирующий действия стаи, как единого целого;

2) в каждый момент t объекты стаи могут быть классифицированы по уровню их активности (ясно, что в общем случае это отношение эквивалентности $\varepsilon(t)$ – нестационарное);

3) каждый предпорядок на множестве объектов стаи, согласованный с отношениями эквивалентности $\varepsilon(t)$, характеризует стратегию выделения лидеров в стае и, следовательно, определяет некоторый механизм преобразования стаи в распределенную систему.

Вероятностный механизм выбора активируемых объектов стаи $\bigcirc = \{o_1, \ldots, o_n\}$ основан на следующих построениях. Пусть $\xi_i(t)$ $(i = 1, \ldots, n)$ – независимые случайные величины, область значений каждой – множество $\{0.1\}$, причем вероятность того, что $\xi_i(t) = 1$ равна $\alpha_i(t) \cdot (\sum_{j=1}^{n} \alpha_j(t))^{-1}$ (содержательно $\xi_i(t) = 1$ означает, что объект $o_i \in \bigcirc$ активируется в момент t). Ясно, что на основе стандартного вероятностного подхода могут быть получены характеристики действий каждого объекта $o_i \in \bigcirc$ $(i = 1, \ldots, n)$, как «в среднем», так и «почти всегда». Отсюда вытекает, что и действия стаи, рассматриваемой как единое целое, могут быть охарактеризованы в терминах «в среднем» и «почти всегда».

Заключение.

Основная цель настоящей работы состоит в выделении основных характеристик управления стаей, т.е. группой не взаимодействующих друг с другом объектов, объединенных общей целью. Показано, что многообразие путей эволюции стаи полностью характеризуется режимами моделирования действий объектов, выбором порядка активации объектов и системой предпочтений на множествах действий объектов. Для задач прикрытия кругового люка крышкой при равномерном распределении объектов на крышке и действиях, представленных композицией параллельного переноса и поворота, а также уничтожения стаей круговой цели, совершающей периодические колебания, эти характеристики были реализованы в компьютерной системе моделирования, построенной на основе Microsoft Visual Studio.NET 2003 TM. Установленная связь между поведением автоматов в лабиринтах и действиями объектов стаи дает возможность с единых позиций охарактеризовать сложность и алгоритмическую разрешимость задач управления стаей и представляет собой предмет дальнейших исследований.

Список литературы

1. Месарович М. и др. Теория иерархических многоуровневых систем. – М.: Мир, 1973.
2. Варшавский В. И. Коллективное поведение автоматов. – М.: Наука, 1973.
3. Cassandras C., Lafortune S. Introduction to Discrete Event Systems. – The Netherlands, Dordrecht: Kluver Academic Publishers, 1999.
4. Luce L. D., Raiffa H. Games and Decisions: Introduction and Critical Survey. – NY: John Wiley, 1957.
5. Bierman H. S., Fernandes L. Game Theory With Economic Applications. – USA, NY: Addison-Wesley Publishing Company, Inc., 1993.
6. Felsenthal D. S., Machover M. The Measurment of Voting Power. – UK, Chelthenham, Edward Elgar Publishing Limited, 1998.
7. Кудрявцев В. Б. и др. О поведении автоматов в лабиринтах // Дискретная математика, Т.4, Вып. 3, 1992. – С. 3-27.
8. Каляев И. А. Стайные принципы управления в группе объектов // Искусственный интеллект, № 3, 2004. – С. 700-714.
9. Скобелев В. Г., Тыкулов Е. В. Стайная модель управления группой объектов // Труды ИПММ НАН Украины, Т. 11., 2005. – С. 126-136.
10. Кудрявцев В. Б. и др. Введение в теорию конечных автоматов. – М.: Наука, 1985.

Распознавание классов лабиринтов буквы A коллективами автоматов

Б. Стаматович, *професcор, biljas@cg.ac.yu*
Математический факультет, Черногория

Изучается проблема существования коллектива автоматов, который распознает класс шахматных лабиринтов. Этот класс, в геометрическом смысле, представляет букву ***A***. В [3] доказано, что для этого класса не существует распознающий автомат. В предлагаемой работе приводится доказательство существования распознающего коллектива автомата типа (1, 1).

Основные понятия и результаты

Основные обозначения и понятия как лабиринт, конечный автомат, коллектив автоматов взяты из [1, 2].

Множество T отрезков на плоскости называется *конфигурацией*, если любые два разных отрезка из этого множества могут иметь не больше одной общей точки, причем если она есть у них, то она обязательно является концевой для обоих отрезков.

Лабиринт L = (V, E), где $V \subseteq R^2$, называем *прямоугольным лабиринтом*, если для любых $u, v \in V$ из $(u, v) \in E$ следует, что отрезок $\overline{uv}$ идет в направлении $|(u, v)|$, а множество отрезков $T = \{\overline{uv} \mid (u, v) \in E\}$ является конфигурацией. Фигура $\overline{L} = \bigcup_{(u,v)\in E(L)} \overline{uv}$ в R^2 называется *реализацией* прямоугольного лабиринта L.

Пусть Z^2 – целочисленная решетка на плоскости. Проведем через вершины Z^2 все возможные прямые, параллельные осям координат. Полученная фигура является реализацией прямоугольного лабиринта, который обозначим через $\mathbf{Z}^2$. Под *мозаичным лабиринтом* будем понимать любую связную часть (нагруженную) лабиринта $\mathbf{Z}^2$. Под *шахматным лабиринтом* будем понимать любой связной подграф (нагруженный) лабиринта $\mathbf{Z}^2$.

Пусть **V** = ((1, -1), (1, 0), (1, 1), (0, -1), (0, 1), (-1, -1), (-1, 0), (-1, 1)) = (p_1, p_2, ..., p_8) упорядоченный набор различных ненулевых элементов из Z^2. Инициальный автомат **A** = (A, Q, B, φ, ψ, q_0) – называется *допустимым*, если A= $P(\{0,1\}^9)\setminus\varnothing$ – множество всех непустых подмножеств множества $\{0,1\}^9$, B = D ∪ {**0**}, где D = { p_2 , p_4 , p_5 , p_7 } и если ψ такое, что для произвольных q ∈ Q и $a = (a_1, ..., a_9) = (1, a_2,..., a_9) \in \{0,1\}^9$ из того, что ψ(q, *a*) = p_i, для некоторого i, $0 \le i \le 8$, следует, что $a_{i+1} = 1$; p_0 = **0** (нуль вектор). На рисунке 1. допускаемые выходы p_5, p_7. Набор **V** называем полем зрения автомата **A** .

В дальнейшем предполагаем, что все автоматы являются допустимыми и все лабиринты шахматным.

Набор **V** определяет для каждого $z \in Z^2$ набор **V**(z) = (z, z + p_1, z + p_2, ..., z + p_8), также и 9-тичный набор в лабиринте L_{v_0}, который обозначим через $[z]_L$. Набор **V**(z) называем окрестностью точки $z \in Z^2$. Поведением автомата $\mathbf{A}_{q_0}$ в лабиринте L_{v_0} называем последовательность $\pi(\mathbf{A}_{q_0}; L_{v_0}) = (q_0, z_0), (q_1, z_1), \ldots$, где $(z_i, z_{i+1}) \in E(L_{v_0})$ или $z_i = z_{i+1}$, $q_{i+1} = \varphi(q_i, [z_i]_L)$ и $\psi(q_i, [z_i]_L) = |z_i, z_{i+1}|$, i = 0, 1, ... Пусть $Int(\mathbf{A}_{q_0}, L_{v_0}) = \bigcup_{i=1}^{\infty} \{z_i\}$. Если $Int(\mathbf{A}_{q_0}, L_{v_0}) = V(L_{v_0})$, то говорим, что автомат $\mathbf{A}_{q_0}$ *обходит* лабиринт L_{v_0} .

0	1	0
1	**1**	0
1	0	1

(**1**,1,0,0,0,1,1,1,0)
рис 1.

Пусть ***A*** = ($\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$), где $\mathbf{A}_i = (A_i, Q_i, B_i, \varphi_i, \psi_i)$, $1 \le i \le n$, *коллектив* автоматов с *полем зрения* **V.** Пусть выделен автомат $\mathbf{A}_i \in \boldsymbol{A}$ удовлетворяющий следующим условиям. $Q_i = \{q_i\}$ и для любого a = (a_1, a_2, ..., a_9) ∈ A_i либо $\psi_i(q_i, a) = 0$, либо если $\psi_i(q_i, a) = b \ne 0$, то существуют s, s ≠ i и q ∈ Q_s, такие, что (s, q)∈ a_1 и ψ_s(q, a') = b, где $a' = \left((a_1 \setminus \{(s,q)\}) \bigcup \{(i, q_i)\}, a_2, ..., a_9\right)$. Тогда автомат $\mathbf{A}_i$ называется *камнем* в коллективе ***A***. Вспомним, это значит что в точке лабиринта, если камень движется, тогда в этой точке существует еще один автомат. Коллектив ***S*** = ($\mathbf{A}_{q_0}$, **K**), где $\mathbf{A}_{q_0}$ не камень, а автомат **K** – камень, называется *коллективом типа (1, 1).*

Будем говорить, что автомат $\mathbf{A}_{q_0}$ (коллектив ***S*** = ($\mathbf{A}_{q_0}$, **K**) типа (1, 1)) *распознает* лабиринт L_v, если при его запуске в лабиринт L_v происходит переход автомата $\mathbf{A}_{q_0}$ в заключительное состояние q_{F_1} , а при его

запуске в лабиринт $L'_{v'} \neq L_v$ происходит переход в заключительное состояние q_{F_0}. Пусть **C** класс инициальных лабиринтов.

Кроме инициального состояния q_0 автомата $A_{q_0} = (A, Q, B, \varphi, \psi, q_0)$, можно ввести *множество заключительных состояний* $Q_F \subseteq Q$. Пусть $Q_F = \{ q_{F_0}, q_{F_1} \}$.

Говорим что автомат $\mathbf{A}_{q_0}$ (коллектив $\boldsymbol{S} = (\mathbf{A}_{q_0}, \mathbf{K})$ типа (1, 1)) *распознает класс* **C** если при его запуске в любой лабиринт L_v происходит переход автомата A_{q_0} в заключительное состояние q_{F_1} только тогда, когда $L_v \in \mathbf{C}$ и для любого лабиринта $L_v \notin \mathbf{C}$ происходит переход в заключительное состояние q_{F_0}.

В [3] определен класс лабиринтов C_A и показано что не существует автомат, который распознает его. Здесь, только неформально опишем этот класс. Элемент из класса C_A можно горизонтальными отрезками разделить на части которые являются элементами класса лабиринтов C_1, C_2 и C_3 (рис 2).

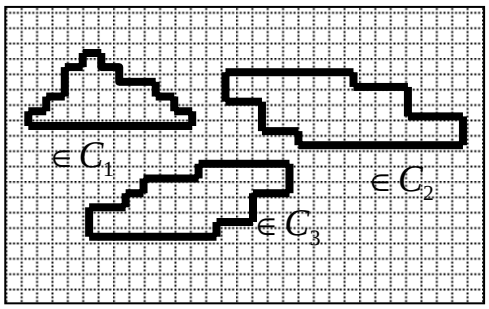


Рис 2.

На рисунке 3. показан элемент из класса C_A. Покажем, что для этого классе существует распознающий коллектив типа (1,1).

Пусть (C_A, v_{hr}) класс инициальных лабиринтов, где стартовая вершина v_{hr} любого лабиринта из класса C_A самая верхняя и самая правая точка в этом лабиринте.

Теорема 1. *Существует коллектив* $\boldsymbol{S} = (\mathbf{A}_{q_0}, \boldsymbol{K})$*, типа (1,1), который распознает класс* (C_A, v_{hr}).

Доказательство .

Заметим что, окрестность точки, в которой автомат находится единственная информация, которую автомат имеет в каждом моменте времени. Автомат не имеет информацию о возможном нахождении в "окрестности" бесконечной внешней области или в окрестности конечной дыры. Следствием этого факта построен лабиринт-ловушка для класса C_A в [3]. Автомат нуждается в дополнительной информации, которая поможет ему при обхождении конечной дыры в одном и том же направлении понять, что он не "зациклился".

Распознающие автоматы $\mathbf{A}_1$, $\mathbf{A}_2$ и $\mathbf{A}_3$ для классов лабиринтов C_1, C_2 и C_3 уже построены в [4]. Здесь, не строим формально автомат. Опишем его обход лабиринта из класса (C_A, v_{hr}). Заметим, что автомат $\mathbf{A}_{q_0}$ движется через лабиринт из класса (C_A, v_{hr}) по частям $K_6, K_4, K_3, K_5, K_2, K_1$, рядом (рис 3.).

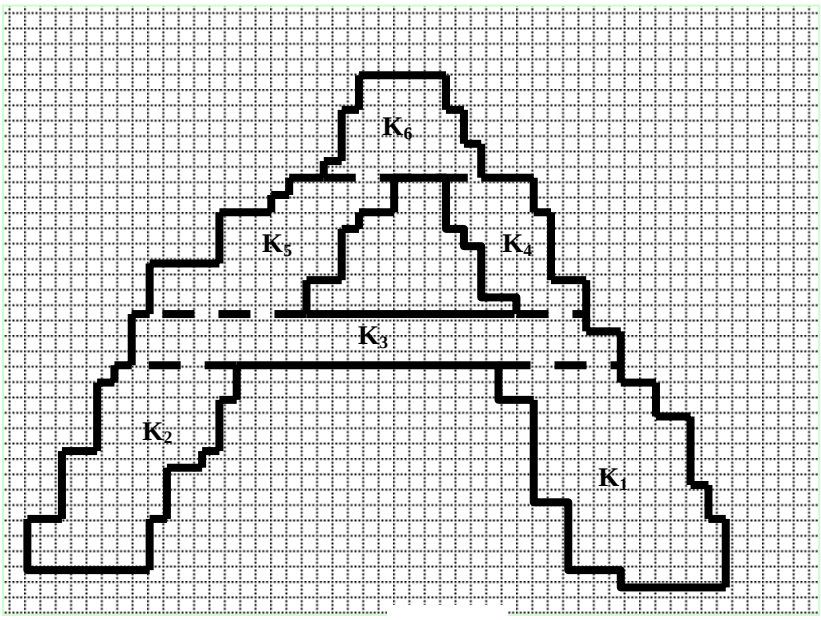


Рис 3.

Автомат $\mathbf{A}_{q_0}$ стартует как автомат $\mathbf{A}_1$, движется налево, вниз, направо, вниз... Камень **K** все время рядом с ним. В некотором моменте коллектив $(\mathbf{A}_{q_0}, \mathbf{K})$ будет в окрестности конечной дыры. Автомат знает, так что находится в точке вида как на рисунке 4., где x∈{0,1}. Камень **K** "остается" в точке

x	x	x
1	**1**	x
0	1	x

рис 4.

лабиринта, которая в окрестности конечной дыры (самая верхняя и самая правая точка дыры) и "помнит", что автомат $\mathbf{A}_{q_0}$ был в этой точке. Автомат $\mathbf{A}_{q_0}$ переходит в множество состояниа которые «управляют» переходом к автомату $\mathbf{A}_3$ и потом ведет себя как автомат $\mathbf{A}_3$. Автомат $\mathbf{A}_{q_0}$ движется по нижней границе дыры. Он переходит в множество состояниа которые «управляют» переходом к автомату $\mathbf{A}_2$ и потом ведет себя как автомат $\mathbf{A}_2$. Автомат $\mathbf{A}_{q_0}$ встретит камень **К**.

Потом, автомат $\mathbf{A}_{q_0}$ движется ниже, по границе дыры, и переходит в множество состояниа которые «управляют» переходом к автомату $\mathbf{A}_1$ и потом ведет себя как автомат $\mathbf{A}_1$. Потом, ведет себя как автомат $\mathbf{A_3}$, а потом ведет себя как автомат $\mathbf{A_2}$.

Автомат $\mathbf{A}_{q_0}$ закончивает свое движение переходом в финальное состоение q_{F_1}.

Если автомат $\mathbf{A}_{q_0}$ запускается в лабиринт, который не принадлежит классу (C_A, v_{hr}), тогда он либо не встретит камень **К**, либо автоматы $\mathbf{A}_1$, $\mathbf{A_2}$, $\mathbf{A_3}$ реагируют переходом в финальное состоение q_{F_0}, либо в множество состояния которые «управляют» переходами между этим автоматами есть финалное состоение q_{F_0}.

Список литературы

1. Килибарда Г. Об обходе конечных лабиринтов системами автоматов // Дискретная математика. – 1990. –Т. 2, вып. 2. – С. 71 – 81
2. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. – М.: Наука, 1985.
3. B. Stamatovic, Automata recognition of a class of chess labyrinths, Mathematica Montisnigri, (accepted)
4. Б. Стаматович, Распознавание двусвязных цифр коллективами автоматов, Интеллектуальные системы, Москва, 1999, Том 3, 321 – 337.

Об одном алгоритме для нахождения приближённого решения задачи о рюкзаке

Сытник А. В.,

Каф. Математической Теории интеллектуальных систем, механико-математического факультета МГУ им. М.В. Ломоносова.

e-mail: mariarty@rbcmail.ru

NP-сложные задачи очень часто возникают в реальности. Для того, чтобы их решать существуют различные подходы. Например, NP-сложные задачи можно решать приближённо с помощью выпуклого программирования, и в частности полуопределённого программирования.

Задача о рюкзаке является одной из наиболее популярных задач в комбинаторной оптимизации. Как известно, это NP - полная задача.

Задача целочисленного программирования с булевыми переменными (0 или 1) называется задачей о рюкзаке, если

$$f(x) = \left[\sum_{i=1}^{n} c_i x_i \right] \to \max_x \quad \text{при} \quad \sum_{i=1}^{n} a_i x_i \le b \,,\ 1 \le i \le n \,,\ x_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Рюкзак загружается предметами n различных типов, предмет $i-$ого типа имеет вес $a_i \ge 0$ и стоимость $c_i \ge 0$. Максимальная грузоподъёмность рюкзака равна $b \ge 0$. Требуется загрузить рюкзак предметами так, чтобы максимизировать функцию полезности f и не превысить допустимую грузоподъёмность. Если $x_i = 0,$ то нет предметов $i-$ого типа, $x_i = 1$ - есть.

Определение. Следующая задача оптимизации называется главной стандартной формой полуопределённого программирования (SDP):

$$\text{минимизировать} \quad C \bullet X$$
$$\text{при условиях} \quad \begin{array}{l} A_1 \bullet X = b_1, \\ \ldots \\ A_m \bullet X = b_m, \\ X \geq 0 \end{array}$$

$C, A_1, \ldots, A_m, X$ – симметричные матрицы $n \times n$. $C, A_1, \ldots, A_m, b_1, \ldots, b_m$ - даны, матрица X – неизвестна. Обозначение $Y \bullet Z$ означает поэлементное внутреннее произведение, т.е. $Y \bullet Z = \sum_{i,j} y_{ij} z_{ij}$.

Теорема. *Существует полиномиальный алгоритм, сложности* $O(n^2)$, *приближённо решающий задачу о рюкзаке, с решением, по крайней мере,* 0.87856 *от оптимума.*

Задача решается путём сведения задачи о рюкзаке к полуопределённой программе, решающейся приблизительно за полиномиальное время.

С помощью данной задачи возможно решать реально существующие задачи, такие как, например: загрузка произвольного транспорта с ограничением по грузоподъёмности.

Автор благодарит академика Кудрявцева В.Б. и доцента Ирматова А. А. за внимание к работе и ценные указания.

Список литературы

[1] Goemans, Williamson – Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming 1995

[2] Goemans, Williamson – New ¾ approximation algorithms for maximum satisfiability problem 1994

[3] Goemans, Williamson – Approximation Algorithms for Max-3-Cut and Other Problems via Complex Semidefinite Programming 2001

[4] Nesterov, Nemerovskii – Interior Point Polynomial Methods in Convex Programming 1994

[5] Nesterov – Introductory Lectures on Convex Programming 1998

[6] Nesterov, Todd – Primal-Dual Interior-Point Methods for Self-Scaled Cones 1998

[7] Alizadeh - Interior point methods in semidefinite programming with applications to combinatorical optimization 1995

[8] Karloff, Zwick – A 7/8 Approximation Algorithm for Max 3Sat 1997

[9] Halperin, Zwick – Approximation Algorithms for MAX-4SAT and rounding procedures for semidefinite programs

[10] Schulman – Probability and Algorithms, Lectures 2003

[11] Krishnan, Mitchell – A semidefinite programming based polyhedral cut and price approach for the maxcut problem 2004

[12] Burer, Monteiro – A Projected Gradient Algorithm For Solving The Maxcut SDP Relaxation 2004

[13] Attalah – Algorithms and Theory of Computation Handbook 1999

[14] Boyd, Vandenberghe – Convex Optimisation 2004

[15] Papadimitriou – Combinatorical Optimisation Algorithms and Complexity 1982

[16] Фролов Андреев Болотов Строгалов – Прикладные задачи дискретной математики в энергетике 1988

[17] Arora, Safra – Probabilistically Checkable Proofs A New Characterization of NP 1998

[18] Yamashita, Fujisawa, Kojima – Semidefinite Programming Algorithm Parallel version 2002

Достижения и проблемы концептуального моделирования

Бернхард Тальхайм,
Институт информатики и прикладной математики, Университет Киля
E-mail: thalheim@is.informatik.uni-kiel.de

Посвящается семидесятилетию проф. В. Б. Кудрявцева

Технологии баз данных и информационных систем претерпели существенные изменения. В настоящее время повсеместное распространение получили системы управления контентом, вебсервисы с интенсивной обработкой информации, сотрудничающие системы, интернет-базы данных, OLAP-базы данных, производящих анализ в реальном времени, и т.п. В то же время благодаря широкому применению технология объектно-реляционных данных стала стабильной и хорошо разработанной. Концептуальное моделирование (пока) не охватило перечисленные выше области. На протяжении десятилетий основным вопросом концептуального моделирования была спецификация структур, тогда как для адекватного представления современных систем необходимо также рассматривать вопросы функциональности, взаимодействия и распределенности. Многие задачи, поставленные еще в 1987 году в работах [13], [14] до сих пор остаются открытыми. В то же время появился целый ряд новых технологий, например объектно-реляционная модели и модели на основе XML. Новые технологии не решили всех проблем, скорее расширили и обострили имевшиеся нерешенные задачи. В работе приводится список открытых задач в классических областях теории баз данных - спецификациях структуры и функциональности. Задачи, связанные с взаимодействием и распределенностью, в настоящее время являются предметом весьма интенсивных исследований.

Постановки открытых задач сопровождается описанием основных результатов в области концептуального моделирования. Представляется подход к моделированию объектно-реляционных сотрудничающих систем с поддержкой виртуальных рабочих групп, интеграции информационных систем, разнообразных архитектур, таких как OLTP-OLAP, play-out и play-in систем и систем анализа данных. Основой работы является расширенная модель отношения элементов (Entity-Relationship, ER), покрывающая все структурные возможности объектно-реляционных систем и использующая теории медиа-типов и сценариев для спецификации взаимодействия.

Введение

Создание и развитие информационных систем

Проблема создания информационных систем может быть сформулирована следующим образом: Для данной СУБД (или парадигмы базы данных) создать физическую и логическую структуру информационной системы так, чтобы система содержала все данные, необходимые для пользователей и для эффективного функционирования системы для всех пользователей. Определить прикладные процессы базы данных и их взаимодействие с пользователями.

Основными задачами создания баз данных являются:

- удовлетворение всех информационных (контекстных) требований для всех пользователей в данной прикладной области;
- реализация "естественной"и простой для понимания структуры хранимой информации;
- обеспечение готовности всей семантической информации для возможных изменений структуры;
- обеспечение требований к обработке данных и высокой эффективности обработки;
- обеспечение логической независимости запросов и транзакций на данном уровне;
- предоставление простых и понятных пользовательских интерфейсов.

В последние годы велось активное обсуждение структур баз данных. Некоторые проблемы были успешно решены. Однако при рассмотрении задачи моделирования возникают новые аспекты:

Структурирование приложений баз данных относительно структуры баз данных и соответствующих статических ограничений целостности.

Функционирование приложений баз данных на основе процессов и динамических ограничений целостности.

Распределение компонент информационной системы, задаваемое явной спецификацией сервисов и интерфейсов взаимодействия.

Интерактивность (взаимодействие с пользователями), определяемое на основе предполагаемых сценариев работы воображаемых пользователей, зависящих от типов носителей, используемых для выдачи информации пользователям и для обработки поступающих данных.

Понимание важности новых аспектов привело к созданию так называемого подхода к соразработке при моделировании на основе спецификаций структуры, функциональности, распределения и интерактивности. Каждый из аспектов имеет как синтаксические, так и семантические элементы.

Однако основную задачу решить так и не удалось.

Открытая задача 1.

Найти общую мотивацию, общую формальную модель и соответствие, удовлетворяющее всем свойствам, и формализовать характеристики.

Общие модели информационных систем

Создание баз данных основывается на одной или нескольких моделях данных. Часто создание сводится исключительно к структурным аспектам. Иногда дополнительно вводится статическая семантика, основанная на статических ограничениях целостности. После реализации структуры происходит спецификация процессов. Поведение процессов специфицируется посредством динамических ограничений целостности. Далее выполняется разработка интерфейсов. Глубина теоретической проработки описанных выше этапов существенно различается, как показано в следующей таблице, представляющей данные на конец 90-х годов.

	Использование на практике	Теоретическая проработка	Начальный уровень спецификации
Структура	хорошо	хорошо проработана	стратегический
Статическая семантика	частично используется	хорошо проработана	концептуальный
Процессы	как-то сделано	отдельные моменты	требования
Динамическая семантика	отдельные части	маленькие куски	реализация
Сервисы	реализации специально для конкретных систем	частные случаи	реализация
Форматы обмена		ничего	реализация
Интерфейсы	интуитивно	ничего	реализация
Сценарии	интуитивно	ничего	реализация

Создание баз данных требует одновременной согласованной разработки структур, процессов, распределений и интерфейсов. Ниже мы покажем, как расширенная модель отношения элементов позволяет работать со всеми четырьмя аспектами.

В настоящее время базы данных расширяются до информационных web-систем, хранилищ данных, интеллектуальных баз знаний и систем анализа данных. Расширение можно проводить консервативным путем или на основе новых парадигм. Консервативный путь является предпочтительным, если только новые парадигмы не позволяют решать бывшие нерешенными задачи. В случае использования консервативного пути необходима хорошая архитектура [8], [6], позволяющая строить расширяемые, масштабируемые системы.

Открытая задача 2.

Найти архитектуру для общего расширения систем баз данных, позволяющую моделировать все службы и делать выводы о свойствах системы.

В то же время необходимо моделировать качество работы систем баз данных. Критерии качества часто задаются весьма нечетко. Распространенными критериями качества являются точность, изменяемость, отказоустойчивость, удобство, производительность, неразглашение, восстановляемость, надежность, эффективность, безопасность, стабильность и верифицируемость [4].

Открытая проблема 3.

Формально определить критерии качества, атрибуты и метрику для оценки качества для концептуального моделирования, а также разработать средства для внедрения, контроля и улучшения качества.

Спецификация структур баз данных

Языки спецификации структур

Структура баз данных основывается на трех взаимозависимых компонентах:

Синтаксис: Индуктивное задание структуры с использованием множества базовых типов, набора конструкторов и теории применения конструкторов, ограничивающей применение конструкторов правилами и формулами деонтической логики. В большинстве случаев теория может быть опущена. Структурная рекурсия является основным средством задания спецификаций.

Семантика: Спецификация допустимых баз данных на основе статических ограничений целостности описывает легальные состояния баз данных. Если используется структурная рекурсия, для описания статических ограничений целостности может использоваться иерархическая логика предикатов первого порядка.

Прагматика: Описание контекста и цели основано либо на явных ссылках на модель фирмы, задачи фирмы, политику фирмы и окружение, либо на интенциональной логике, задающей интерпретации и значения в зависимости от времени, местонахождения и здравого смысла.

Индуктивное задание структуры основано на базовых типах и конструкторах типов. Базовый тип представляет собой алгебраическую структуру $B = (Dom(B), Op(B), Pred(B))$ с именем, областью допустимых значений, множеством операций и множеством предикатов. Класс B^C базового типа представляет собой набор элементов из $dom(B)$. Как правило требуется, чтобы B^C было множеством. Также B^C может быть списком, мультимножеством, деревом и т.п. Классы могут изменяться под воздействием операций. Элементы классов могут классифицироваться с помощью предикатов.

Конструктор типов представляет собой функцию из множества типов в новые типы. В состав конструктора может входить оператор выбора для извлечения данных (например Select) и операторы обновления данных (например Insert, Delete, Update) для отображения из нового типа в компоненты типов или в новый тип. Операторы обновления могут быть нагружены критериями корректности результата, правилами верификации, подразумеваемыми правилами, пользовательскими представлениями, физическими представлениями или свойствами физического представления.

В качестве примера типичных для баз данных конструкторов можно привести конструкторы множеств, n-компонентных векторов, списков и мультимножеств. Конструктор множеств основан на некотором определенном типе и использует алгебру операций, включающую объединение, пересечение и дополнение. Можно считать, что в этом случае оператор выбора принимает в качестве аргумента предикат. Операторы обновления, такие как Insert или Delete, определяются как выражения в алгебре множеств. В пользовательском представлении используются фигурные скобки { и }. Конструкторы типов определяют систему типов над базовыми схемами данных, т.е. набор конструируемых множеств значений данных. В некоторых моделях баз данных конструкторы типов основываются на семантике указателей.

Именование и ссылки также являются удобными средствами конструирования в моделях. Каждый тип и класс концепции имеет имя. Эти имена могут быть использованы для для определения новых типов; на имена можно ссылаться при определении типа. Часто структуры включают необязательные компоненты. Необязательные компоненты и ссылки следует использовать с максимальной осторожностью, так как в противном случае потребуется использовать сверхсложные логики, такие как логики топов [9]. Более удачным подходом к моделированию является требование слабой идентифицируемости по значениям всех объектов баз данных [10].

Ограничения целостности

Ограничения целостности используются для отделения "хороших"состояний или последовательностей состояний системы баз данных от нежелательных состояний или последовательностей. Ограничения целостности используются для спецификации как семантики, так и процессов в базах данных. Следовательно непротиворечивость приложений баз данных не может рассматриваться отдельно от ограничений. В то же время ограничения целостности задаются пользователем на различных уровнях абстракции, с различными видами неопределенности и подтекстами, на разных языках. Для обработки и практического использования, однако, ограничения должны быть

определены явно и однозначно. Для устранения этого противоречия пользовательские ограничения транслируются во внутрисистемные процедуры, реализующие поддержание целостности.

В каждой структуре также имеется множество подразумеваемых наследуемых из модели ограничений целостности.

- Ограничения конструирования компонент основаны на существовании, мощности и включении компонент. Такие ограничения должны учитываться в процессе трансляции и импликации.

- Ограничения идентифицируемости неявно используются в конструкторе множеств. Каждый объект должен либо не принадлежать множеству, либо входить в множество ровно один раз. Множества основаны на простых общих функциях. Свойство идентифицируемости, однако, представимо только в терминах автоморфизмов групп [1]. Позднее мы увидим, что представимость значений и слабая представимость значений влекут за собой контролируемость структуры.

- Ацикличность и конечность структуры поддерживают аксиоматизируемость и определенность алгебры. Данные ограничения необходимо выражать явно. Ограничения типа ограничений на мощность могут основываться на потенциально бесконечных циклах.

- Внешнее структурирование означает представимость ограничений посредством структуры. В этом случае сложно характеризовать импликации ограничений.

Подразумеваемые наследуемые из модели ограничения целостности относятся к фильтрам производительности и поддержания.

Ограничения целостности могут быть сформулированы в терминах BV-форм (Beeri-Vardi frames), т.е. импликаций, в левой и правой и правой части которой стоят формулы. BV-ограничения, вообще говоря, не задают строгое ограничение выразимости. Если структура является иерархической, BV-ограничения могут быть заданы в рамках логики предикатов первого порядка. Можно ввести целый ряд классов ограничений целостности, таких как:

- Ограничения для построения равенств позволяют для множества объектов из одного или нескольких классах генерировать равенства самих объектов или их компонент.

- Ограничения для построения объектов требуют, чтобы для множества объектов, удовлетворяющих некоторому условию, существовало другое множество объектов.

Класс $\mathcal{C}$ ограничений целостности называется замкнутым относительно импликации Гильберта, если он может быть аксиоматизирован конечным множеством ограниченных правил вывода и конечным множеством аксиом. Известно, что множество зависимостей объединений не является замкнутым относительно импликации Гильберта для реляционных структур. Однако существует аксиоматизация с неограниченным правилом, т.е. правилом с потенциально бесконечным числом посылок.

Основной проблемой является извлечение ограничений целостности. Так как необходимо обрабатывать множества, возникает необходимость в более сложных теориях вывода. Хорошим кандидатом является визуальный или графический вывод, гораздо более сильный, чем логический вывод [3].

Открытая задача 4.

Создать средство вывода для обработки множеств ограничений. Классифицировать "реальные"множества ограничений, которые могут быть легко заданы и поддерживаемы.

Еще несколько проблем, связанных с зависимостями, могут быть сформулирована на уровне реляционной модели. Приведем соответствующие постановки.

Открытая задача 5.

Является ли проблема импликаций для зависимостей замыкания и функциональных зависимостей алгоритмически разрешимой? Является ли эта проблема аксиоматизируемой?

Какие подклассы ограничений по включению, содержащие унарные ограничения по включению, аксиоматизируемы вместе с классом функциональных зависимостей?

Какие подклассы зависимостей объединения, содержащие класс многозначных зависимостей, являются аксиоматизируемыми?

Охарактеризовать отношения, совместимые по функциональным зависимостям.

Охарактеризовать свойства классов ограничений при горизонтальной декомпозиции.

Способы представлений

Классический подход к объектам баз данных заключается в хранении объектов в зависимости от их типов. Таким образом, каждая сущность оказывается представленной группой объектов, объединенных либо с помощью идентификаторов, либо с помощью специальных поддерживающих процедур. Однако, вообще говоря, следует рассматривать два различных подхода к представлению объектов:

- Поклассовое представление на основе идентификаторов: Хранимые в базе данных сущности могут представляться несколькими объектами. Идентификаторы объектов поддерживают идентификацию без рассмотрения сложных взаимоотношений между объектами. Объекты могут быть элементами нескольких классов. На раннем этапе развития объектного подхода предполагалось, что такой класс единственный. Подобное допущение приводило к ряду проблем при миграции объектов, не имевших удовлетворительного решения.

  Описанное выше представление используется в структурах на основе расширенных ER- моделей [14] и объектно-ориентированных моделей. В реляционных и объектно-реляционных системах баз данных используется следующий подход:

- Пообъектное представление: В графовых моделях, разработанных для облегчения применения объектного подхода [1], объекты представлены в виде подграфов, т.е. в виде совокупности вершин, ассоциированных с объектом, и соответствующих ребер. Такое представление соответствует представлению, используемому в процессе стандартизации.

  XML основан на пообъектном представлении. Он позволяет использовать нулевые значения без уведомления. Если значение объекта не существует, неизвестно, неприменимо, не может быть получено и т.п., XML-схема не использует тэг, соответствующий атрибуту или компоненте. Классы являются скрытыми.

Пообъектное представление вносит существенную избыточность, которая должна поддерживаться системой, таким образом существенно снижая производительность. Помимо производительности, проблемами также являются низкая масштабируемость и неэффективное использование ресурсов. Работа с подобными системами приводит к лавинообразному появлению блокировок — любая модификация данных требует рекурсивной блокировки связанных друг с другом объектов.

Суммируя вышесказанное, пообъектное представление применимо только при выполнении следующих условий:

- Приложение изменяется мало, а структура данных и поддерживающие основные функции приложения не изменяются на протяжении жизненного цикла системы.

- Обновления данных производятся очень редко. Модификация, вставка и удаление разрешаются только внутри четко определенных ограниченных "зон"базы данных.

Типичными примерами областей применения систем с пообъектным представлением являются архивы, системы представления информации и системы управления контентом. Такие системы строятся поверх подсистемы обновления данных и называются play-out системами. Данные хранятся в том же виде, в котором они передаются пользователю. Подсистема модификации данных содержит play-out-генератор, генерирующий все необходимые просмотры данных для play-out системы.

Другим примером являются базы данных без обновлений, такие как в системе SAP, содержащей огромное число взаимозависимых просмотров.

Первое представление может быть использовано для средств поиска, второе — для ввода и вывода данных в хранилищах данных.

Открытая задача 6.

Построить технику и теорию для обработки избыточных множеств объектов с поддержкой управления целостностью множеств объектов, например наборов XML-документов.

Оптимизация баз данных основывается на знании сложности операций. Если известно, что некоторое подмножество операций существенно более сложное, чем остальные операции, и известно несколько эквивалентных представлений, среди таких представлений можно найти наиболее простое. Примером оптимизации является вертикальная нормализация, ориентированная на разложение отношений на множество отношений, имеющих меньшую сложность и более простых в поддержании. Горизонтальная нормализация ориентирована на выбор частей отношений с наименьшей сложностью. Дедуктивная нормализация ориентирована на сведение базы данных к отношениям, которые не могут быть получены из других отношений применением правил вывода. Напомним, что при нормализации получающиеся представления должны оставаться эквивалентным исходному. В настоящее время описанные виды нормализации рассматриваются по-отдельности.

Открытая задача 7.

Найти общую модель для применения вертикальной, горизонтальной и дедуктивной нормализации для объектно-реляционных моделей данных.

Нормализация часто основывается на ограничениях баз данных.Для проведения корректной нормализации необходимо знать все множество ограничений на данные для рассматриваемого приложения. Но эта задача слишком сложна и часто принципиально нерешаема.

Открытая задача 8.

Найти теорию нормализации, применимую в случае неполноты множества ограничений.

Спецификация функциональности

Операции в информационных системах

Общие операции над системами типов могут быть определены с помощью структурной рекурсии. Пусть заданы типы T и T', множественный тип C^T над T (например, множество значений типа T, мультимножество, список) и операции, такие как обобщенное объединение $\cup_{C^T}$, обобщенное пересечение $\cap_{C^T}$ и обобщенный пустой элемент $\emptyset_{C^T}$. Пусть также задан элемент h_0 типа T' и две функции $h_1 : T \to T'$ и $h_2 : T' \times T' \to T'$. Тогда операция структурной рекурсии представления вставки R^C над T определяется следующим образом.

$$srec_{h_0,h_1,h_2}(\emptyset_{C^T}) = h_0$$

$$srec_{h_0,h_1,h_2}(|\{|s|\}|) = h_1(s) \text{для одноэлементных наборов } |\{|s|\}|$$

$$srec_{h_0,h_1,h_2}(|\{|s|\}| \cup_{C^T} R^C) = h_2(h_1(s), srec_{h_0,h_1,h_2}(R^C)), \text{если } |\{|s|\}| \cap_{C^T} R^C = \emptyset_{C^T}.$$

Все операции объектно-реляционной модели, ER-модели, и других декларативных моделей баз данных могут быть определены в терминах структурной рекурсии, например

- выбор определяется операцией $srec_{\emptyset,i_\alpha,\cup}$, где

$$i_\alpha = \begin{cases} \{o\}, & \text{если } \{o\} \models \alpha \\ \emptyset & \text{в противном случае} \end{cases}$$

- функция агрегирования может быть определена на основе двух функций для нулевых значений

$$h_f^0(s) = \begin{cases} 0, & \text{если } s = NULL \\ f(s) & \text{в противном случае} \end{cases}$$

$$h_f^{undef}(s) = \begin{cases} undef, & \text{если } s = NULL \\ f(s) & \text{в противном случае} \end{cases}$$

и структурной рекурсии, например

$$sum_0^{null} = srec_{0,h_{Id}^0,+} \text{или } sum_{undef}^{null} = srec_{0,h_{Id}^{undef},+};$$

$$count_1^{null} = srec_{0,h_1^0,+} \text{или } count_1^{undef} = srec_{0,h_1^{undef},+}$$

или с помощью SQL-определения, например функция вычисления среднего значения может быть представлена как $sum_0^{null}/count_1^{null}$.

Аналогично можно определить пересечение, объединение, разность, проекцию, соединение, вложение, переименование, вставку, удаление и обновление.

Выразительная сила структурной рекурсии также ограничена. Недетерминированные программы, генерирующие вектора или объекты, не могут быть выражены в терминах структурной рекурсии.

Операции могут использоваться либо для извлечения значений, либо для изменения состояния базы данных.

Для определения операций используется подход, основанный на просмотрах, предназначенных для ограничения области значений, а также предусловий и постусловий, ограничивающих применимость, активизируемых операций, а также явного описания принудительно выполняемых операций: Операция φ

[Просмотр: <Имя_просмотра>]

[Предусловие: <Условие_активации>]

[Активизируемая_операция <Спецификация>]

[Постусловие: <Условие_принятия>]
[Принудительные_операции: <Операция, Условие>]

Реляционная модель, а также объектно-реляционные модели могут быть расширены за счет добавления операций агрегирования, группировки и ограниченной рекурсии. Семантика перечисленных операций варьируется от СУБД к СУБД и до сих пор не получила математического обоснования [5].

Открытая задача 9.

Разработать общую теорию операций, расширяющих объектно-реляционные модели.

Динамические ограничения целостности

Динамика функционирования баз данных определяется посредством систем переходов. Система переходов для схемы S представляет собой пару $\mathcal{TS} = (\mathcal{S}, \{\xrightarrow{a} | a \in \mathcal{L}\})$ где $\mathcal{S}$ — непустое множество переменных состояния, $\mathcal{L}$ — непустое множество (меток), $\xrightarrow{a} \subseteq \mathcal{S} \times (\mathcal{S} \cup \{\infty\})$ для каждой метки $a \in \mathcal{L}$. Переменные состояния задают состояния. Переходы представляют собой транзакции в S.

Время жизни базы данных задается в терминах путей в $\mathcal{TS}$. Путь π в системе переходов представляет собой конечную или счетную последовательность вида $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots$. Длина пути есть число переходов.

В системе переходов $\mathcal{TS}$ можно ввести темпоральную динамическую логику баз данных, используя кванторы $\forall_f$, (всегда в будущем), $\forall_p$ (всегда в прошлом), $\exists_f$ (в некоторый момент в будущем) и $\exists_p$ (в некоторый момент в прошлом).

Логика первого порядка может быть расширена за счет темпоральных операторов. Функция истинности расширяется за счет рассмотрения времени. Пусть имеется темпоральный класс (R^C, l_R). Функция истинности I расширяется за счет рассмотрения времени и определяется на $S(t_S, R^C, l_R)$. Формула α истинна для $I_{(R^C, l_R)}$ в момент t_S, если она истинна для базы данных в момент t_S, т.е. $I_{(R^C, l_R)}(\alpha, t_S) = 1$ тогда и только тогда, когда истинна $I_{S(t_S, R^C, l_R)}(\alpha, t_S)$.

- Для формул без темпоральных префиксов расширенная истинность совпадает с обычной истинностью.
- $I(\forall_f \alpha, t_S) = 1$ тогда и только тогда, когда $I(\alpha, t_S) = 1$ для всех $t'_S > t_S$;
- $I(\forall_p \alpha, t_S) = 1$ тогда и только тогда, когда $I(\alpha, t_S) = 1$ для всех $t'_S < t_S$;
- $I(\exists_f \alpha, t_S) = 1$ тогда и только тогда, когда $I(\alpha, t_S) = 1$ для некоторого $t'_S > t_S$;
- $I(\exists_p \alpha, t_S) = 1$ тогда и только тогда, когда $I(\alpha, t_S) = 1$ для некоторого $t'_S < t_S$.

Модальные операторы $\forall_p$ и $\exists_p$ (и $\forall_f$ и $\exists_f$) являются двойственными, т.е. формулы $\forall_h \alpha$ и $\nexists_h \alpha$ эквивалентны. С помощью следующих правил описанные операторы могут быть отображены в классическую модальную логику:

$$\Box\alpha \equiv (\forall_f \alpha \wedge \forall_p \alpha \wedge \alpha);$$

$$\Diamond\alpha \equiv (\exists_f \alpha \vee \exists_p \alpha \vee \alpha).$$

Можно дополнительно ввести темпоральные операторы until и next.

Наиболее важным классом динамических ограничений целостности являются ограничения на переход $\alpha O \beta$, задающие предусловие α и постусловие β для каждой операции O. Ограничение $\alpha O \beta$ могут быть выражено темпоральной формулой $\alpha \xrightarrow{O} \beta$.

Произвольное конечное множество статических ограничений целостности может быть эквивалентным образом записано в виде множества ограничений на переход $\{\Lambda_{\alpha \in \Sigma^\alpha} \xrightarrow{O} \Lambda_{\alpha \in \Sigma^\alpha} | O \in Alg(M)\}$.

Ограничения целостности могут накладываться:

- на процедурном уровне, с помощью:
  - введения триггеров [7] в так называемые активные настройки событие-условие-действие;
  - задания максимальных операторов, не нарушающих целостности [9];
  - хранимых процедур, т.е. программ, определяющих все возможные нарушения ограничений целостности.
- на уровне транзакций, ограничивая последовательности переходов из состояния в состояние последовательностями, не нарушающими ограничения целостности;

- на уровне СУБД, на основе декларативных спецификаций, зависящих от возможностей СУБД;

- на уровне интерфейса, путем рассмотрения операций, не нарушающих целостность.

Ограничения на базы данных отображаются в ограничения на переходы. Ограничения на переходы хорошо изучены, прежде всего благодаря их локальности. Такие ограничения могут поддерживаться с помощью триггеров или хранимых процедур. Однако вопрос глобальных взаимозависимостей остается открытым.

Открытая задача 10.

Разработать теорию взаимного влияния ограничений целостности баз данных, отображаемого на удобные множества триггеров и хранимых процедур.

Задание последовательности выполняемых действий

В литературе предлагалось значительное количество подходов к заданию последовательности выполняемых действий. С нашей точки зрения предпочтительным является формальное описание с графическим представлением, позволяющее избегать проблем, связанных с методами чисто графического задания, такими как и/или ловушки. Рассмотрим алгебру базовых шагов вычисления, введенную в [16].

- Базовыми управляющими командами являются последовательное выполнение ; (выполнение шагов по очереди), распараллеливание $|\wedge|$ (выполнение шагов в параллельном режиме), исключающий выбор $\oplus$ (выбор одного пути выполнения из нескольких возможных), синхронизация $|sync|$ (синхронизация двух параллельных путей выполнения с помощью синхронизирующего условия), и простое слияние + (слияние двух параллельных путей выполнения). Исключающий выбор является подразумеваемой параллельной операцией и обозначается $||$.

- Структурными управляющими командами являются произвольные циклы $*$ (выполнение шагов без каких-либо структурных ограничений на циклы), произвольные циклы + (выполнение шагов без каких-либо структурных ограничений на циклы, за исключением того, что цикл должн быть пройден по крайней мере один раз), опциональное выполнение [] (выполнение шага один раз или пропуск шага), неявное прерывание $\downarrow$ (закончить выполнение, если больше не осталось шагов), вход в подшаг $nearrow$ и завершение подшага $searrow$.

Расширим алгебру с помощью дополнительного набора команд.

- Дополнительные команды ветвления и синхронизации включают в себя множественный выбор $|(m,n)|$ (выбрать из всего множества возможных путей выполнения от m до n путей), множественное слияние (слияние нескольких путей выполнения без синхронизации), дискриминатор (слияние нескольких путей выполнения без синхронизации с выполнением последующих шагов не более одного раза), объединение n из m (слияние нескольких путей выполнения с частичной синхронизацией и выполнением следующего шага не более одного раза), и синхронизирующее объединение (слияние нескольких путей выполнения; если путей несколько, производится синхронизация, в противном случае выполняется простое слияние).

- Можно также рассмотреть управляющие команды на множестве объектов (CMO-команды), такие как CMO-команды с получением информации на этапе проектирования (сгенерировать множество реализаций одного шага, причем мощность множества определяется на этапе проектирования), CMO-команды с получением информации на этапе выполнения (сгенерировать множество реализаций одного шага, причем мощность множества определяется в некоторый момент выполнения, как, например, в FOR-циклах), CMO-команды без получения предварительной информации (сгенерировать множество реализаций одного шага, причем мощность множества неизвестна, как, например, в WHILE-циклах), и CMO-команды с синхронизацией (синхронизирующие ребра, создать множество реализаций одного действия и провести синхронизацию после его выполнения).

- Управляющие команды на основе состояний включают отклоняющий выбор (выполнить один из двух путей, причем выбирается подразумеваемый путь), наложенное параллельное выполнение (выполнить два действия в случайном порядке, но не в параллель, а последовательно), и предел (выполнение действий вплоть до достижения некоторого предела).

- Отменяющие команды включают в себя отмену шага, отмену ветви и т.п.

Описанные выше операторы являются обобщениями шаблонов последовательностей выполняемых действий, созданными в соответствии с подходами, разработанными для алгебр сетей Петри.

Операции, определенные с помощью описанной идеологии, могут быть непосредственно оттранслированы в программы для баз данных. На данный момент не существует теории поведения баз данных, которая могла бы охватить поведение баз данных во всей широте и глубине. Отправной точкой для создание такой теории, возможно, станет изложенное в работе [15] предложение использовать абстрактные машины состояний [2].

Открытая задача 11.

Разработать теорию поведения баз данных. Теория должна охватывать работу как самой базы данных, так и системы управления базой данных.

Архитектура СУБД

Функционирование информационных систем моделируется с помощью декомпозиции множества состояний системы на четыре типа состояний:

$$\begin{aligned}\mathcal{E}R^C = \quad & (\text{входные состояния } \mathcal{IN}, \text{выходные состояния } \mathcal{OUT}, \\ & \text{состояния СУБД } \mathcal{DBMS}, \text{состояния базы данных } \mathcal{DB})\end{aligned}$$

Входные состояния охватывают входную информацию системы, т.е. запросы и данные. Выходные состояния отражают вывод СУБД, т.е. выходную информацию и сообщения об ошибках. Состояния СУБД содержат внутренние состояния системы управления. Состояния базы данных описывают содержимое базы. Все четыре класса состояний могут быть структурированы. Например, если состояние базы данных структурируются в соответствии со схемой данных, входные состояния структурируются аналогично.

При использовании ориентированных на значения или объектно-реляционных моделей состояния баз данных могут быть представлены отношениями. В этом случае обновление одного из типов схемы отображается в изменение одного из отношений. Изменения состояний моделируются с помощью правил изменения состояний абстрактных машин состояний [2]. СУБД задается программой и управлением. Под программой мы будем понимать элементы работы или сервисов, удовлетворяющие заданным критериям качества, под управлением и координацией — манипулирование блоками программ, возможно с дополнительными требованиями атомарности и непротиворечивости. Также управление и координация могут задаваться с помощью команд управления задачами.

При вызове программ производится инициализация значений входных переменных. Переменные могут быть статическими, храниться на стеке, задаваться явно или неявно. Дополнительно могут использоваться такие параметры вызовов, как onSubmit (ввод значения) и presentationMode (режим презентации), параметры приоритета onFocus (фокусировка на процессе) и emphasisMode (режим выделения), параметры управления onRecovery (восстановление после сбоя) и hookOnProcess (активация процесса), параметры ошибок onError (рассмотрение ошибок) и notifyMode (режим уведомлений), а также общие параметры передачи, такие как onReceive (активация приема) и validUntil (ограничение времени жизни значения).

Требования атомарности и непротиворечивости поддерживаются в рамках ряда моделей транзакций. В качестве примера можно привести плоские транзакции, сага-транзакции, контракты и т.п. [14].

Пусть T' — подтип СУБД $\mathcal{E}R^C$. Рассмотрим изменение $T(s_1, \ldots, s_n) := t$. Множество $\mathcal{U} = \{T_i(s_{i,1}, \ldots, s_{i,n_i}) := o_i | 1 \leq i \leq m\}$ объектно-ориентированных изменений состояний называется непротиворечивым, если для всех $1 \leq i < j \leq m$ из равенства $T_i(s_{i,1}, \ldots, s_{i,n_i} = T_j(s_{j,1}, \ldots, s_{j,n_j}$ следует равенство $o_i = o_j$.

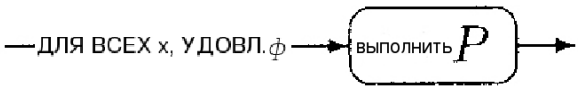
Результатом выполнения непротиворечивого множества $\mathcal{U}$ изменений состояний из $\mathcal{E}R^C$ является новое состояние $\mathcal{E}R^C + \mathcal{U}$, для каждого объекта o из $\mathcal{E}R^C$ определяемое по формуле

$$(\mathcal{E}R^C + \mathcal{U})(o) = \begin{cases} Update(T_i, s_{i,1}, \ldots, s_{i,n_i}, o_i), \\ \text{ если } T_i(s_{i,1}, \ldots, s_{i,n_i}) := o_i \in \mathcal{U} \\ \mathcal{E}R^C(o), \text{ в противном случае.} \end{cases}$$

Параметризованная программа $r(x_1, \ldots, x_n) = P$ арности n состоит из имени r, правила перехода P и множества свободных переменных $\{x_1, \ldots, x_n\}$ правила P.

Информационная система $\mathcal{E}R^C$ является моделью формулы ϕ ($\mathcal{E}R^C \models \phi$), если $[[\phi]]_\zeta^{\mathcal{E}R^C} =$ истина для всех возможных значений ζ свободных переменных ϕ.

Рассмотрим две типичных конструкции для порождения программ. Выполнение программы для всех значений, удовлетворяющих некоторому условию, может быть представлена следующей схемой:



Выполнение программы как шага цикла может быть представлено следующей схемой:

Дополнительно рассмотрим такие конструкторы, как последовательное выполнение, ветвление, параллельное выполнение, выполнение после присваивания значений, выполнение после выбора произвольного значения, переход на следующий шаг, изменение состояния информационной системы и вызов подпрограммы.

Используем абстрактные машины состояний и для определения семантики программ.

Правило перехода $\mathcal{P}$ порождает множество изменений состояний $\mathcal{U}$ в состоянии $\mathcal{E}R^C$, если $\mathcal{U}$ непротиворечиво. Происходит изменение состояния информационной системы с присваиванием значений переменных ζ, обозначаемое $yields(\mathcal{P}, \mathcal{E}R^C, \zeta, \mathcal{U})$.

Семантика правил перехода определяется с помощью исчисления правил следующего вида:

$$\frac{\text{предусловие}_1, \ldots, \text{предусловие}_n}{\text{вывод}} \text{где условие.}$$

Например, изменение состояния, заданное первым из описанных конструкторов, определяется правилом

$$\frac{\forall a \in I : yields(\mathcal{P}, \mathcal{E}R^C, \zeta[x \mapsto a], \mathcal{U}_a)}{yields(\text{для всех } x, \text{удовлетворяющих } \phi, \text{выполнить } \mathcal{P}, \mathcal{E}R^C, \zeta, \cup_{a \in I}\mathcal{U}_a)}$$

$$\text{где } I = range(x, \phi, \mathcal{E}R^C, \zeta).$$

Диапазон $range(x, \phi, \mathcal{E}R^C, \zeta)$ представляет собой множество $\{o \in \mathcal{E}R^C | [[\phi]]^{\mathcal{E}R^C}_{\zeta[x \to a]} = \text{истина}\}$.

Открытая задача 12.

Разработать общую теорию абстракций и уточнений для систем баз данных, поддерживающую различные архитектуры и позволяющую декомпозировать базы данных на компоненты.

Задание распределения

Задача задания распределение долгое время игнорировалась. Явное задание распределения не применялось, вместо этого использовались различные подходы моделирования сотрудничающих систем, такие как системы нескольких баз данных или союзничающие системы баз данных.

Комплект просмотров

По классическому определению, (простой) просмотр представляет собой одноэлементный тип, данные которого выбираются из базы по некоторому запросу вида

создать просмотр <имя>	(<переменные, на которые проецируется просмотр>)
выбрать	<выражение, задающее проекцию>
из	<подсхема базы данных>
где	<условие выбора>
сгруппировать по	<выражение для группировки>
с	<выбор из групп>
упорядочить по	<порядок выдачи информации в просмотре>

Так как возможно использование поклассового представления, простые просмотры могут оказаться неоптимальной структурой для спецификации обмена информацией. Предпочтительной моделью является комплект просмотров. Комплект состоит из множества элементов, схемы интеграции или ассоциирования элементов и требований к поддержанию отношения ассоциирования.

Простые примеры комплектов просмотров рассмотрены в работе [14], где в качестве комплектов выступают ER-схемы. Интеграция задается схемой. Требования к поддержанию основаны на

концепции "главный-подчиненный", т.е. состояние классов комплекта просмотров изменяется при изменении соответствующих областей базы данных.

Просмотры должны также реализовывать поддержку сервисов. Сервисы предоставляют свои данные и функциональность. Подобная объектная ориентированность удобна, если возникает необходимость использования данных без установления прямого или удаленного соединения с СУБД.

Рассмотрим обобщенную форму задания просмотра для реляционных баз данных:
сгенерировать <отображение: переменные → выходная структура>
из <типы базы данных>
где <условие выбора>
представление с использованием <общий стиль представления>
& <абстракция (гранулярность, мера, точность)>
& <упорядочение в рамках представления>
& <иерархические представления>
& <точки просмотра>
& <разделения>
с учетом определений <условие>
& <перемещение>
с использованием функций <функции поиска>
& <функции экспорта данных>
& <функции ввода данных>
& <функции сессии>
& <функции разметки>

Расширение просмотров за счет функций кажется избыточным на этапе проектирования баз данных. Использование просмотров в распределенных средах удается минимизировать усилия, связанные с параллельной и последовательной разработкой, так как сразу генерируется комплект просмотров вместо того, чтобы отдельно разрабатывать каждый просмотр.

Сервисы

Традиционно сервисы изучались в рамках одного из (семи) уровней коммуникационных систем и характеризовались двумя параметрами: функциональностью и качеством обслуживания. Однако мы используем более современный подход [8] и будем рассматривать не функции, а информационные процессы. Качество обслуживания ограничивается рядом свойств, формулируемых или на уровне реализации, или на уровне концепции, или на уровне пользователей. Сервис состоит из информационного процесса, предоставляемых характеристик и свойств, гарантирующих качество обслуживания. Формально сервис представляет собой тройку $\mathcal{S} = (\mathcal{I}, \mathcal{F}, \Sigma_{\mathcal{S}})$, где $\mathcal{I} = (\mathcal{V}, \mathcal{M}, \Sigma_{\mathcal{T}})$.

Информационный процесс задается тремя компонентами:

- Просмотры из комплекта $\mathcal{V}$ являются ресурсами информационного процесса. Так как просмотры расширены за счет функций, они обладают вычислительными возможностями и могут использоваться в качестве статистических пакетов, хранилищ данных или средств раскопок данных.

- Менеджер сервиса $\mathcal{M}$ поддерживает функциональность и качество обслуживания и управляет контейнерами, их play-out- функциями и доставкой информации клиентам. Менеджеры сервисов также называются провайдерами сервисов.

- Область применимости сервиса задается в виде множества допустимых задач $\mathcal{T}$.

Характеристики сервиса $\mathcal{F}$ задаются в зависимости от уровня абстракции.

- Характеристики на уровне пользователей основаны на соглашениях уровня сервиса и информационных процессах этого уровня.

- Характеристики на уровне концепции описывают свойства, которыми сервис должен обладать для выполнения соглашений уровня сервиса. Здесь же задаются доступные клиентам функции путем спецификации интерфейсов и семантики.

- Характеристики на уровне реализации описывают синтаксические интерфейсы функций, предоставляемые данные, поведение и ограничения на информационную систему и клиентов.

Качество обслуживания $\Sigma_{\mathcal{S}}$ задается в зависимости от уровня абстракции.

- Параметры качества на уровне пользователей включают всеохватность (неограниченность доступа в пространстве и времени), и безопасность (по отношению к сбоям, атакам, ошибкам; степень доверенности).

- Параметры качества на уровне концепции включают интероперабельность (формальный каркас для интерпретации) и непротиворечивость (функций данных).

- Параметры качества на уровне реализации включают долговечность (доступ ко всей информации, если не оговорено противное), надежность (на основе моделей сбоев для устойчивости, конфликтов и живучести), производительность (на основе модели расходов, времени отклика и пропускной способности), и масштабируемость (по отношению к изменениям сервисов, числа клиентов и серверов).

Формы обмена информацией

Форма обмена информацией определяется следующими параметрами.

- Архитектура обмена обычно основывается на архитектуре системы, интегрирующей информационные системы с помощью подсистем коммуникаций и обмена.

- Стиль сотрудничества задает поддерживающие программы, стиль взаимодействия и координирующие средства.

- Шаблон сотрудничества задает роли партнеров, права и обязанности и протоколы общения.

Распределенные системы баз данных основываются на локальных системах баз данных, объединенных с помощью заданной стратегией интеграции. Основой интеграции является полное объединение локальных концептуальных схем в глобальную схему распределения. Архитектура модели представлена на рис. 1.

Помимо классических распределенных систем можно также рассмотреть и другие архитектуры, такие как фермы баз данных, наращиваемые сообщества информационных систем и сотрудничающие информационные системы. Последняя модель основана на концепции сотрудничающих просмотров [14]. Наращиваемые сообщества информационных систем являются основой систем управления оборудованием. Простыми примерами таких систем являются хранилища данных и средства управления контентом.

Фермы баз данных являются обобщением и расширением подхода, заложенного в союзничающие информационные системы и системы-посредники. Архитектура ферм баз данных изображена на рис. 2. Фермы основаны на подходе к соразработке и концепциях информационного элемента и контейнера.

- Информационные элементы — это обобщенные просмотры. Просмотры генерируются на основе содержимого базы данных. Элементы — это просмотры, функциональность которых расширена, чтобы использовать собственные данные. Информационные элементы могут быть ориентированы либо на извлечение, либо на модификацию данных. Элементы первого типа используются для введения новых данных, элементы второго типа — для модификации локальных бах данных.

- Контейнеры поддерживают экспорт и импорт данных с помощью информационных элементов, предоставленных состояниями просмотров. Элементы объединяются в контейнеры, которые могут быть подгружены или выгружены специальным образом. Процедура выгрузки поддерживает диалоговые сцены и шаги.

- Система глобальных коммуникаций и ферм предоставляет протоколы обмена, средства загрузки и выгрузки контейнеров и средства модификации элементов данных, ориентированных на модификацию.

Задача полного объединения локальных баз данных не ставится. Задача заключается в создании сотрудничающих просмотров.

Архитектура обмена может включать в себя рабочие места клиентов, описывающие актеров, группы, роли и права актеров в рамках групп, пакет задач и организацию сотрудничества, коммуникации и взаимодействие.

Стиль сотрудничества определяется четырьмя компонентами:

- поддерживающими программами информационных систем, включающими управление сессиями, управление пользователями и систему тарификации и оплаты;

Рис. 1. Обобщение трехуровневой архитектуры на случай распределенной схемы

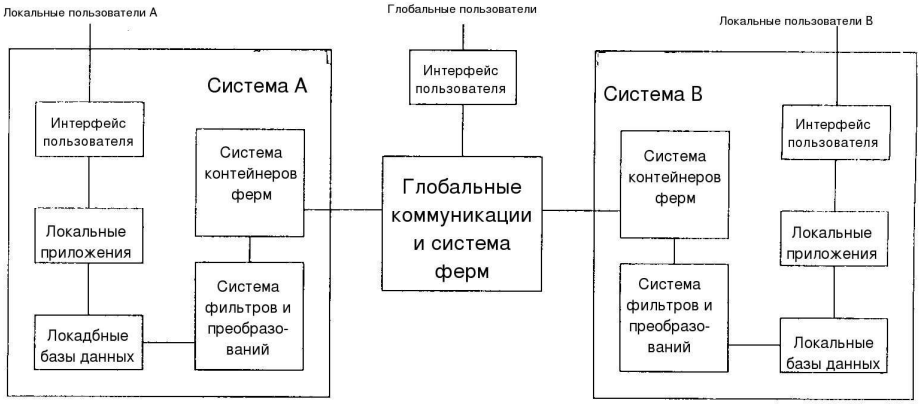


Рис. 2. Ферма баз данных

- шаблонами доступа для передачи данных через сеть, например многоадресная или одноадресная передача, для разделения ресурсов, на основе либо моделей транзакций, согласия и восстановления, либо репликации с управлением сбоями, и для удаленного доступа, включая планировщик доступа;

- стилем взаимодействия на основе одноранговых моделей или компонентных моделей или моделей инициирования событий с ограничением на возможные коммуникации;

- координацией последовательностей выполняемых действий, описывающей сотрудничество партнеров, типы диалогов, отображение пространств имен и правила сотрудничества.

Известен целый ряд шаблонов сотрудничества, поддерживающих доступ и конфигурирование (оборачивающая фронтальная компонента, конфигурирование компонент, перехватчик, расширяющие интерфейсы), обработку событий (пререагирование, постреагирование, асинхронные токены завершения, соединение приема), синхронизацию (блокирование подобластей, блокирование с заданной стратегией, интерфейсы безопасного взаимодействия многонитевых программ, оптимизация блокировок с двойной проверкой) и параллельное выполнение (активные объекты, мониторинг, полусинхронное полуасинхронное выполнение, ведущий/ведомый, определяемое нитями хранение).

- Сотрудничество на основе проксирования использует частичные копии системы (удаленный прокси-сервер, прокси-сервер защиты, кэширующий прокси-сервер, синхронизирующий прокси-сервер и т.д.).

- Сотрудничество на основе посредников поддерживает координацию коммуникаций, осуществляемую напрямую, через передачу сообщений, на основе концепций торговли, с помощью систем адаптеров-посредников, или с помощью систем отзыва посредников.

- Сотрудничество на основе отношения главный/подчиненный использует жесткое реплицирование данных в соответствии с различными прикладными сценариями (отказоустойчивость, параллельное выполнение, улучшение точности; реплицирование процессов или нитей; реплицирование с координацией или без координации).

- Сотрудничество на основе отношения клиент/координатор основано на пространствах имен и их отображении.

- Сотрудничество на основе отношения издатель/подписчик также иногда называется концепцией зависимости от наблюдателя. Могут рассматриваться как активные, так и пассивные подписчики. У каждого подписчика имеется профиль подписки.

- Сотрудничество на основе отношения модель/просмотр/управление аналогично трехуровневой архитектуре систем баз данных. Просмотры и управление задают интерфейсы.

Шаблоны сотрудничества обобщают протоколы за счет включение в рассмотрение партнеров по процессу, их права, ответственности и роли.

Спецификация интерактивности

Интерактивность информационных систем как правило рассматривается на уровне систем представления, путем архитектурного или Seeheim-разделения системы приложений и системы представления. Структура и функциональность задаются в терминах языка моделирования баз данных и соответствующей алгебры. Прагматика как правило не рассматривается в рамках моделей баз данных. Интерактивность по отношению к системе приложений основывается на множестве просмотров, определенных на структуре базы данных и поддерживаемых некоторой функциональной базой.

Общая архитектура информационной web-системы представлена на рис. 3. Данная архитектура успешно применялась в более чем 30 проектах по созданию огромных или очень больший web-сайтов с интенсивной информационной загрузкой и в более чем 100 проектах по созданию больших информационных систем.

В рамках подхода к соразработке данный подход обобщен за счет следующих добавлений:

- введение новых типов объектов (медиа-объектов), по сути являющихся обобщенными просмотрами, расширенными с помощью соответствующей функциональности, адаптированными к нуждам пользователей и доставляемых актерам с помощью контейнеров [11];

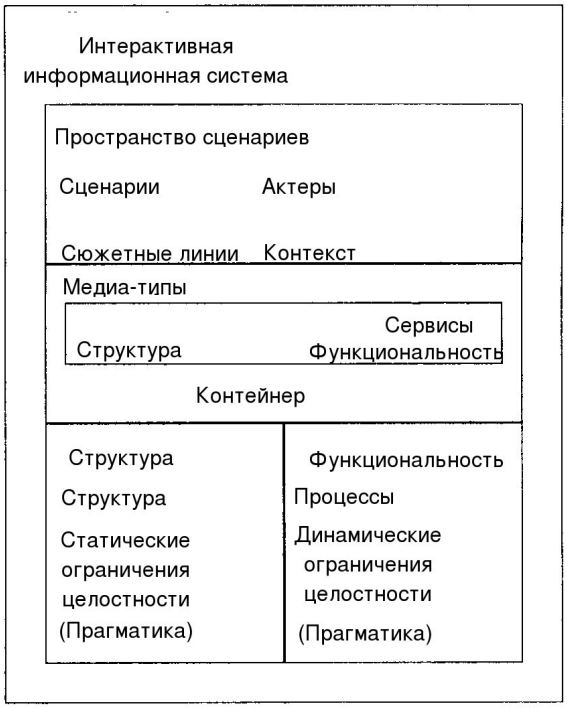


Рис. 3. Подход к спецификации информационных систем

- введение пространств сценариев [12], задающих сценарии использования ресурсов различными группами пользователей (называемых актерами) в рамках некоторого контекста. Сценарии могут генерироваться на основе реальных действий и с помощью различных play-out-средств.

Модель взаимодействия с пользователями включает нескольких партнеров (сгруппированных по некоторым признакам; представители групп называются актерами), рассматривает разнообразные действия и описывает взаимозависимости действий. Помимо последовательности шагов взаимодействия, контента шагов взаимодействия и формы взаимодействия модель охватывает также окружение системы, задачи и актеров.

Пространство сценариев

Модели взаимодействия должны поддерживать множество сценариев. При этом необходимо учитывать профили и окружение пользователей. Сценарий взаимодействия представляет собой сюжет рассказа о работе пользователя или перечень событий. Язык SiteLang [16] предоставляет средства для определения пространств сценариев, сцен и сюжетных линий в рамках сценариев.

В рамках сценария можно выделить нити активности, называемые сюжетными линиями, т.е. пути, составленные из сцен и переходов между сценами. Пространство сценариев есть семерка $\Sigma_W = (S_W, T_W, E_W, G_W, A_W, \lambda_W, \kappa_W)$, где S_W, T_W, E_W, G_W и A_W — множество сцен, созданных W, множество переходов, множество возможных событий, множество средств защиты и множество действий W, соответственно. Таким образом, T_W является подмножеством $S_W \times S_W$. Далее, $\lambda_W : S_W \to SceneSpec$ — функция, ассоциирующая с каждой сценой ее спецификацию, а $\kappa_W : T_W \to E_W \times G_W \times A_W$, $t \mapsto (e, g, a)$ — функция, ассоциирующая с каждым переходом t событие e, инициирующего переход t, средство защиты g (т.е. логическое условие, блокирующее переход в случае ложности при событии e) и действие a, выполняемое при переходе.

Сцены представляют собой точки, в которых происходит взаимодействие, т.е. диалог. Диалоги могут задаваться с помощью так называемых выражений шагов диалога. Каждая сцена обладает уникальным идентификатором Идентификатор_Сцены. С каждой сценой ассоциируется медиаобъект, множество вовлеченных актеров, спецификация представления и контекст. Таким образом, сцена представляет собой следующую структуру:

Сцена = (Идентификатор_Сцены
Выражение шагов диалога
Пользователь
Идентификатор_Пользователя

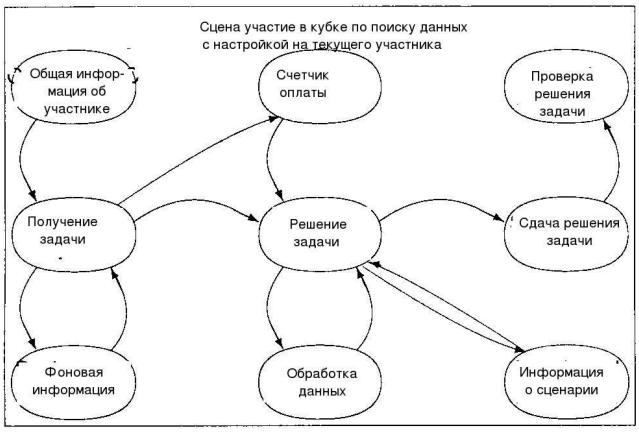


Рис. 4. Одна из сцен для активного обучения

Права_Пользователя
Задачи_Пользователя
Роли_Пользователя
Представление (стили, подразумеваемые значения, ...)
Контекст (оборудование, канал, подробности).

Выражение шагов диалога состоит из диалогов и примененных к диалогам операторов. Типичная сцена представлена на рис. 4. Участник соревнования по поиску данных может ввести свои решения. Для участия в соревновании необходимо внести организационный взнос. Система может знать, а может и не знать пользователя и его профиль. Если пользователь уже внес организационный взнос, диалог оплаты не выводится. Если же пользователь не внес взнос или неизвестен системе, диалог ввода решений доступен только после успешного окончания диалога оплаты.

Комплект медиа-типов

Медиа-типы были введены в работе [11]. Так как разным пользователям в зависимости от истории работы, профиля и окружения требуются существенно отличающиеся данные, пакеты данных передаются через контейнеры. Контейнеры обладают всей функциональностью комплектов просмотров. Комплекты медиа-типов основаны на комплектах просмотров, снабженных специальными средствами доставки и извлечения. Комплекты медиа-типов управляются системой, состоящей из трех компонент:

- Система извлечения медиа-объектов: Медиа-объекты извлекаются и удаляются из базы данных или базы знаний, обобщаются и встраиваются в другие медиа-объекты.

- Система хранения и поиска медиа-объектов: Медиа-объекты могут генерироваться налету, когда требуется обратиться к их содержимому, или храниться в подсистеме хранения и поиска. Так как порождение медиа-объектов как правило имеет высокую сложность, и для нормального функционирование требуется поддержание нескольких версий, предпочтительным способом является хранение.

- Система доставки медиа-объектов: Медиа-объекты используются в разнообразных задачах, разнообразными пользователями в различных социальных и организационных контекстах, в разнообразных окружениях. Система доставки медиа-объектов осуществляет доставку медиа-объектов пользователям в требуемой форме. Контейнеры содержат и управляют множеством медиа-объектов, доставляемых одному пользователю. Пользователь получает адаптированный под него контейнер и может использовать этот контейнер в своей локальной базе данных.

Описанный подход очень близок к концепции хранилищ данных. Он также основан на классической концепции модель-просмотр-управление. Концепция обобщена на медиа-объекты, которые могут просматриваться различными способами, могут генерироваться и управляться генераторами.

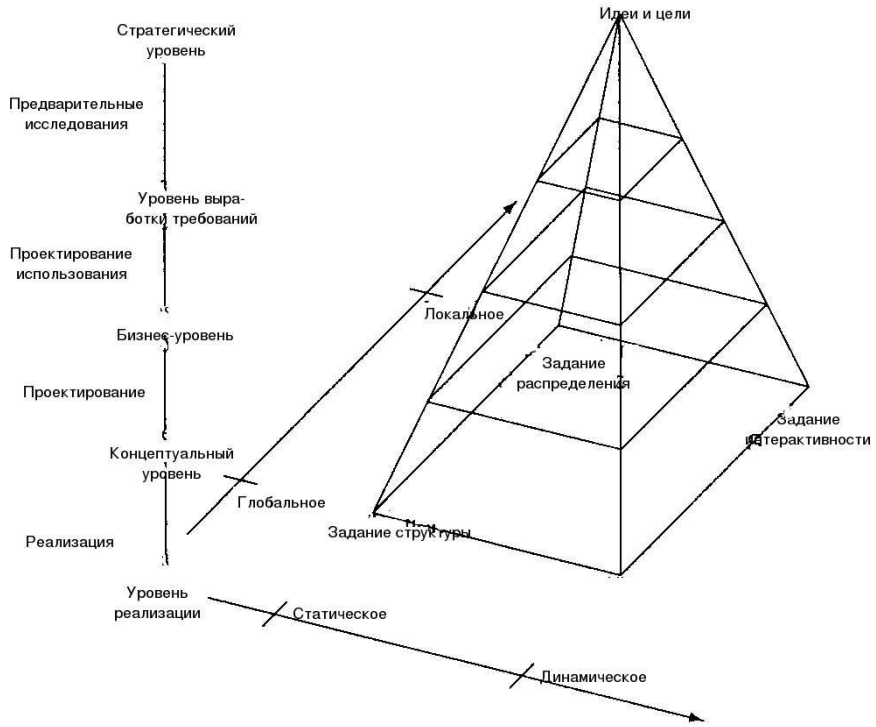


Рис. 5. Модель уровней абстракции процесса разработки баз данных

Интегрирование спецификационных аспектов в соразработку

Описанные выше языки довольно сложны, а непротиворечивая разработка всех аспектов информационных систем трудна. Мы разработали ряд методологий, позволяющих обойти ряд трудностей, связанных с непротиворечивой и полной разработкой. Основой методологий является проектирование сверху вниз (поэтапное уточнение), разделяющее интересующие аспекты на несколько уровней абстракции и использующее операции расширения, детализации, реструктурирования и уточнения.

Модель уровней абстракции для разработки информационных систем

Информационная система может быть задана на разных уровнях абстракции (рис. 5):

1. Стратегический уровень моделирует цели информационной системы, т.е. основную задачу и предполагаемые типы клиентов и решаемые ими задачи. Результатом проектирования на стратегическом уровне является спецификация организационного контракта.

2. Уровень выработки требований описывает информационную систему, анализирует бизнес-процессы и цели для формулирования требований к информационной системе. Результатом проектирования на уровне выработки требований является спецификация системы.

3. Бизнес-уровень моделирует предполагаемое использование информационной системы в терминах типов клиентов, месторасположения пространств информации, переходов между пространствами, а также диалогов между пользователями различных категорий (называемых актерами). Результатом проектирования на уровне бизнес-процессов является расширенное руководство по системе, включающее макеты интерфейсов и сценарии использования.

4. Концептуальный уровень ориентирован на интегрирование концептуальных спецификаций структуры, функциональности, распределения и интерактивности. Результатами проектирования на концептуальном уровне являются схема базы данных, последовательности выполняемых действий, комплекты просмотров и медиа-типов, спецификация сервисов и форматов обмена, а также сценарии.

5. Уровень реализации ориентирован на спецификацию логических и физических структур баз данных, процедур поддержания целостности, программ и интерфейсов. Спецификация осуществляется в рамках языков выбранной платформы. Результатом проектирования на уровне

реализации является модель реализации. Модель существенно зависит от создателя информационной системы.

6. Уровень использования здесь не рассматривается. Поддержка, обучение, введение и администрирование как правило находятся вне сферы концептуального моделирования приложений.

Методология соразработки

Методологии должны соответствовать требованиям к непротиворечивой разработке систем SPICE версии 2.0 и SW-CMM версии 2.0. Соразработка основана на пошаговом уточнении системы при движении по уровням абстракции. Так как четыре аспекта информационных систем — структура, функциональность, распределение и интерактивность — взаимозависимы, они не могут разрабатываться отдельно друг от друга. Приведенная ниже методология основана на шагах следующей структуры:

Правило #i **Имя шага**	Задача 1. Задача 2. ...
Используемые документы	Документы предыдущих шагов (документы по разработке системы) Документация и информация от клиентов
Изменяемые документы	Документы по разработке системы Контракты
Задачи и цели	Общие задачи шага Согласованные цели шага Основная цель
Вовлеченные актеры	Актер A, например представители клиента Актер B, например разработчик
Теоретические основы	Теория баз данных Теория организации Информатика Теория познания, психология, педагогика
Методы и эвристики	Синтаксис и прагматика Используемые языки спецификаций Подходы к упрощению
Разработанные документы Результаты	Документы по разработке системы Результаты, в том числе поставляемые клиенту
Условия начала шага	Условия наличия информации, выполняемые на стороне клиентов Условия зависимости информации Условия на участие
Условия завершения шага	Полнота и правильность критериев Подписанные бумаги, контракты Критерии качества Выполнение задач шага

Используются следующие шаги:

Стратегический уровень

1. Разработка видения, целей и задач

2. Анализ проблем и конкурентов

Уровень выработки требований

3. Разбиение на компоненты

4. Создание наброска пространства сценариев

5. Создание наброска комплекта просмотров
6. Задание бизнес-процессов

Бизнес-уровень

7. Разработка сюжетных линий в пространстве сценариев
8. Выявление основных типов данных и ассоциаций между ними
9. Разработка ядра ограничений целостности, например ограничений идентификации
10. Задание действий пользователей, требований используемости, и создание наброска медиа-типов
11. Выявление требований всеохватности и безопасности

Концептуальный уровень

12. Задание пространства сценариев
13. Разработка типов данных, ограничений целостности, их реализации
14. Задание комплекта просмотров, сервисов и форматов обмена
15. Проектирование последовательностей выполняемых действий
16. Контроль результатов на тестовых данных, тестовых процессах и тестовых сюжетных линиях.
17. Задание комплекта медиа-типов
18. Модулярное уточнение типов, просмотров, операций, сервисов и сцен
19. Нормализация структуры
20. Интеграция компонент по всей архитектуре

Требования реализации

21. Преобразование концептуальных схем в логические схемы, программы и интерфейсы
22. Разработка логических сервисов и форматов обмена
23. Разработка решений для повышение производительности, настройка
24. Преобразование логических схем в физические схемы
25. Проверка долговечности, надежности, масштабируемости и расширяемости

Методология соразработки использовалась на практике для реализации большого количества информационных систем и вместе с тем имеет надежную теоретическую основу. Нашей задачей является не конкуренция с UML, а поддержка разработки систем на твердой основе, без неясностей, пропусков и несочетаемых концепций.

Заключение. Исследования баз данных и информационных систем привели к созданию технологии, ставшей частью современной инфраструктуры. Базы данных используются как встроенные системы, например в автомобильных навигационных программах, как совокупность сотрудничающих систем и как одиночные системы. Технология достаточно хорошо отработана для того, чтобы устанавливать системы баз данных в любых приложениях с поддержкой вычислений, основанных на обработке большого объема информации. Современные информационные системы часто реализуют поддержку распределенных вычислений и web-технологии. Новые архитектуры подняли новые проблемы, которые в настоящее время являются предметом интенсивных исследований. В работе показано, как классическая теория баз данных может быть расширена для решения описанных задач. Отметим, что предлагаемый подход является одним из возможных, могут применяться и другие решения.

В то же время в области исследования баз данных остаются и открытые задачи. В работе приведен обзор современных достижений в области теории баз данных, в основном связанных с вопросами структуры и функциональности. Некоторые результаты уже неприменимы к новым моделям

информационных систем. Однако остается возможность встраивания классических компонент в новые модели. В первой части работы приведен обзор основных результатов и постановки открытых задач.

Список литературы

1. Beeri, C., and Thalheim, B. "Identification as a primitive of datavase models". In Proc. Fundamentals of Information Systems, 7th Int. Workshop on Foundations of Models and Languages for Data and Objects — FoMLaDO'98 (Timmel, Ost-friesland, 1999), T. Polle, T. Ripke, and K.-D. Schewe, Eds., Kluwer, London, pp. 19–36.

2. Borger, E., and Stark, R. "Abstract state machines — A method for high-level design and analysis."Springer, Berlin, 2003.

3. Demetrovics, J., Molnar, A., and Thalheim, B. "Graphical and spread-sheet reasoning for sets of functional dependencies."In ER'2004 (2004), LNCS 3255, pp. 54–66.

4. Jaakkola, H. "Software quality and life cycles."In ADBIS'05 (Tallinn, September 2005), Springer.

5. Lenz, H.-J., and Thalheim, B. "Olap databases and aggregation functions."In 13th SSDBM 2001 (2001), pp. 91–100.

6. Lenz, H.-J., and Thalheim, B. "OLTP-OLAP schemes for sound applications."In TEAA 2005 (Trondheim, 2005), vol. LNCS 3888, Springer, pp. 99–113.

7. Levene, M., and Loizou, G. "A guided tour of relational databases and beyond."Springer, Berlin, 1999.

8. Lockermann, P. "Information system architectures: From art to science."In Proc. BTW'2003, Springer, Berlin (2003), pp. 1–27.

9. Schewe, K.-D. "The specification of data-intensive application systems."PhD thesis, Brandenburg University of Technology at Cottbus, Faculty of Mathematics, Natural Sciences and Computer Science, 1994. Advanced PhD Thesis.

10. Schewe, K.-D., and Thalheim, B. "Fundamental concepts of object oriented databases."Acta Cybernetica 11, 4 (1993), 49–81.

11. Schewe, K.-D., and Thalheim, B. "Modeling interaction and media objects."In NLDB. Natural Language Processing and Information Systems, 5th Int. Conf. on Applications of Natural Language to Information Systems, NLDB 2000, Versailles, France, Jun 28–30, 2000, Revised Papers (2001), M. Bouzeghoub, Z. Kedad, and E. Metais, Eds., vol. 1959 of LNCS, Springer, pp. 313–324.

12. Srinivass, S. "A calculus of fixpoints for characterizing interactive behavior of information systems."PhD thesis, Brandenburg University of Technology at Cottbus, Faculty of Mathematics, Natural Sciences and Computer Science, 2001.

13. Thalheim, B. "Open problems in relational database theory."Bull. EATCS 32 (1987), 336–337.

14. Thalheim, B. "Entity-relationship modeling - Foundations of database technology."Springer, Berlin, 2000. See also `http://www.is.informatik.uni-kiel.de/-thalheim/HERM.htm.`

15. Thalheim, B. "ASM specification of internet information services."In Proc. Eu-rocast 2001, Las Palmas (2001), pp. 301–304.

16. Thalheim, B., and Dusterhoft, A. "Sitelang: Conceptual modeling of internet sites."In ER (2001), H. S. Kunii, S. Jajodia, and A. S01vberg, Eds., vol. 2224 of LNCS, Springer, pp. 179–192.

Замечание: Основной задачей работы было проведение обзора текущего состояния исследований в области баз данных. В библиографию включены только источники, на которые есть ссылки в работе. Обширная библиография по тематике работы содержится, например, в [14].

Моделирование запоминания элементарных математических фактов с помощью нейронных сетей[1]

Татузов А. Л.,
доктор технических наук
ведущий научный сотрудник, НИИ ПМС
124617, Москва, Зеленоград, 1457, 57.
E-mail: itatuzov@yandex.ru

При построении систем искусственного интеллекта, также как и для понимания существа интеллекта естественного, необходимо уметь описывать, хотя бы приблизительно, формирование абстрактных понятий, и умение ими оперировать.

Теория нейронных сетей, широко и успешно применяющаяся в настоящее время для создания систем искусственного интеллекта в самых разнообразных приложениях, использует численные значения. Первоначально зародившись как обобщение попыток понять работу биологических нейронных сетей, искусственные нейронные сети, в основном, опираются на работу с числовой информации. Даже, если решаемая задача требует использования символьных или других нечисловых данных, эти данные преобразуются в числа, а затем работает стандартная схема обработки. Это не всегда разумно. Во-первых, преобразование в числовое представление зачастую обедняет используемую информацию или привносит в нее неоправданное смещение, затрудняя получение наиболее эффективных решений. Во-вторых, понимание работы мозга или биологических нейронных сетей (а ведь именно в этом состояла первоначальная задача в этой области) с помощью современных моделей крайне затруднительно.

В связи с этим важно уяснить, каким образом формируются и развиваются абстрактные понятия у человека. Задача эта исключительно сложна, и в настоящее время имеются лишь самые общие подходы к ее решению. Само понятие абстракции, возникающей в коре головного мозга человека, трудно определить. Дополнительные сложности возникают в связи с затруднительностью проверки имеющихся гипотез из-за скудости достоверного экспериментального материала.

В связи с вышеуказанным, представляется целесообразным на первых этапах ограничиться некоторым более-менее узким классом понятий, которые, тем не менее, описывают мыслительные способности. Интересным в этом контексте является исследование форм представления и использования мозгом натуральных чисел. Видимо, способность считать (оперировать числами, а не только различать количество предметов) имеется только у человека и высших обезьян. Наиболее простой операцией, в которой проявляется свойство собственно абстракции, – это запоминание математических фактов, таких, как таблица умножения.

Таблица умножения имеет законченный вид, допускает удобное представление в компьютерных программах, качество ее усвоения может быть достаточно просто проверено. Ее объем не очень велик, и, тем не менее, существует немало людей, которые допускают ошибки. В отличие от других вариантов фактов математические факты являются абстрактными и, чаще всего, не несут явных прямых связей с огромным потоком информации, в котором живет человек. Поэтому их запоминание можно с определенной степенью точности считать изолированным от остальных знаний человека и рассматривать задачу обучения в "чистом" виде.

Одним из первых начал изучать механизмы запоминания таблицы умножения проф. Андерсона (Anderson J.). Его нейросетевая архитектура Brain-State-in-a-Box как раз и появилась как модель запоминания таблицы умножения. Возможность построения нейросетевых моделей для моделирования запоминания человеком математических фактов исследовалась и другими группами ученых (Edelman B., Abdi H., McCloskey M. Cohen N., Lindemann A.). Достигнуто хорошее соответствие функционирования разработанных моделей с результатами тестирования групп испытуемых. При этом, правда, оценивалось совпадение не с конкретным испытуемым, а с суммарным результатом воспоминания математических фактов всеми испытуемыми, что не совсем корректно. Помимо этого, имеется целый ряд трудностей, включая реализацию в моделях вероятностной структуры ответов, реализации большего числа ошибок при умножении на "трудные" числа ("7", "8") и другие. Основные свойства запоминания таблицы умножения, которые вызывают трудности при нейросетевой имитации, следующие.

1. Наличие случайной составляющей в ответах, человек может давать разные ответы на один и тот же пример, в то время как большинство нейросетевых моделей, как и любые компьютерные программы, выдают всегда один и тот же результат.

[1] Исследования выполнены при поддержке РГНФ (грант № 06-06-00328а).

2. Явно выраженная неоднородность в качестве обучения при изменении величин сомножителей.

Необходимо отметить удивительную скудость доступного экспериментального материала. Несмотря на то, что потенциальный объем экспериментальных данных очень велик, большинство исследователей ссылается на одни и те же результаты. При этом они описывают характеристики запоминания не каждого испытуемого, а всей совокупности в целом. Поэтому и нейросетевые модели сравниваются не с механизмами конкретного индивидуума, а с обобщенными результатами, что не совсем корректно. Этот недостаток может быть преодолен посредством построения многих однородных моделей и их настройкой на каждого из них.

Второй момент заключается в необходимости анализа не статической, уже сформировавшейся структуры памяти изучаемых фактов, а изучении динамики ее формирования в ходе учебного процесса. При этом можно проверить не только совпадение итогов запоминания фактов анализируемой моделью и реальными учениками, но и определить правдоподобность предположений об источниках возникновения ошибок, то есть, более точно понять способ формирования ассоциативных связей в мозге.

Предлагается для изучения свойств запоминания таблицы умножения человеком использовать результаты учебной деятельности школьников. В последние годы широкое развитие получили методы компьютерного обучения, а таблица умножения является идеальным предметом для этого. Получаемые и фиксируемые в ходе компьютерного обучения результаты могут служить великолепной базой для проверки различных нейросетевых моделей. Причем, данные в этом случае будут описывать не только конечные результаты изучения таблицы, но и всю динамику процесса обучения, позволяя сравнивать промежуточные результаты и корректировать модели. Дополнительным стимулом к развитию модели запоминания математических фактов на примере таблицы умножения является возможность использования таблицы умножения в реальном учебном процессе.

Используется следующая модель запоминания (рис. 1, 2). Общая схема нейронной сети представляет собой двухслойный персептрон. Общее число входов нейронной сети для сомножителя составляет 10 для амплитудного представления и столько же для дополнительного паттерна. Общее число входов равно 40. Выходом сети являются два разряда, каждый из которых имеет аналогичное входам представление. Ответ формируется в виде суперпозиции выходов для каждого разряда. Причем имеет место дополнительная ассоциативная связь, направленная от старших разрядов к младшим. То есть, если, например, ответ начинается на "двадцать", то предпочтительными оказываются продолжения "1", "4", "5", "7" и "8", из них "4" с наибольшим приоритетом. Следуя Д.Андерсону, каждое число представляется в «размытом» виде. Наиболее перспективным следует считать формирование понятия числа в виде аттракторной сети в некотором нейронном модуле. Нейроны, активные при поступлении на вход близких чисел, определяют степень «размытости».

Передача сигналов в сети осуществляется посредством активации соответствующих аттракторов (характерных паттернов возбужденных нейронов), что схоже современными представлениями о работе мозга.

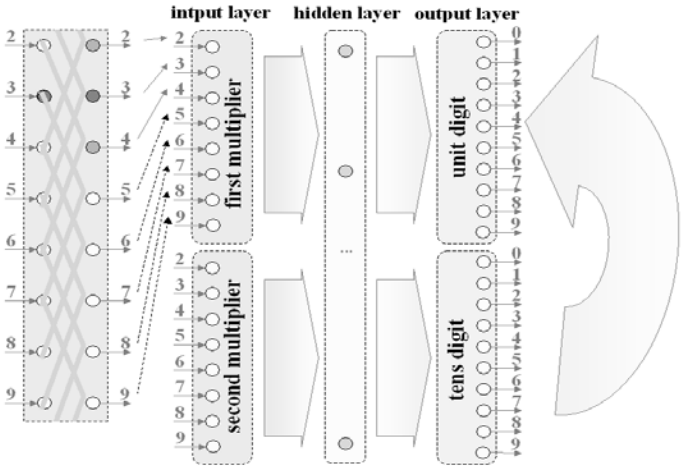


Рис.1. Предварительная схема нейронной сети запоминания математических фактов.

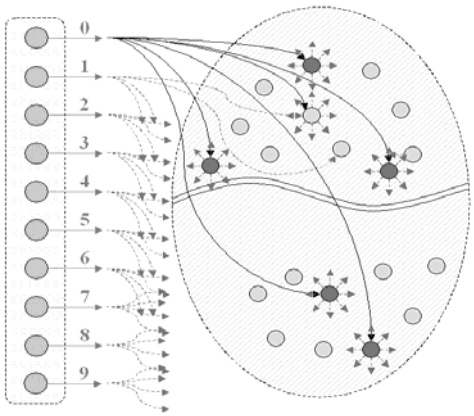


Рис.2. Модуль представления числа в виде аттракторной сети.

Выбор оптимальных наборов обучающих примеров можно осуществлять несколькими способами.

Наиболее простой заключается в предварительном полном обучении сети до получения уверенных правильных ответов во всех примерах. Затем сеть переобучается на примерах, в которых ученик допускает ошибки, стремясь давать те же ошибочные ответы. После этого, отыскиваются наборы

примеров, посредством обучения которым, сеть сможет вновь обучиться правильным ответам. Такой достаточно долгий путь отыскания обучающих примеров можно сократить, взяв все те примеры, которые сеть не сможет правильно выполнять после разучивания, включая ошибки ученика и дополнительно возникшие в ходе переобучения ошибки. Более сложный, но и более соответствующий решаемым задачам состоит в последовательной настройке нейросетевой модели на каждый ответ ученика.

Для изучения влияния способа обучения на результативность усвоения материала исследования проводились с двумя группами, в каждой из которых было примерно одинаковое количество детей с высоким и низким уровнем развития. Группы обучались различными методами. Первой группе для повторения предлагались только задания, в которых на предыдущих упражнениях допускались ошибки, дети второй группы в дополнение получали случайные примеры, третья группа обучалась с помощью примеров выбранных путем нейросетевой оптимизации. Результаты обучения приведены на рисунке 3.

Оказалось, что нейросетевая модель весьма полезна в процессе обучения. Группа, в которой обучающие примеры подбирались нейросетевыми методами, показала уровень усвоения материала на 10…20 % выше, чем у остальных учеников.

Получены только предварительные результаты, однако, они свидетельствуют о перспективности предложенного подхода. В последующих исследованиях предполагается рассмотреть более глубокие нейросетевые модели с использованием более подробного моделирования механизмов ассоциативного запоминания, увеличить объем экспериментов, изучить динамику процесса запоминания.

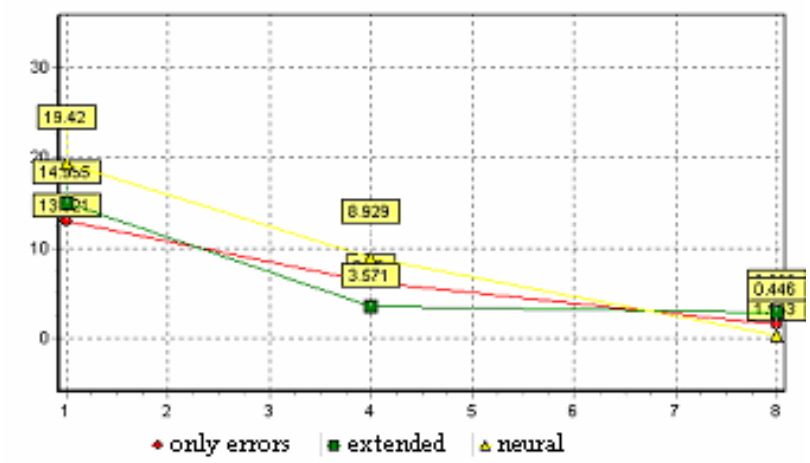


Рис.3. Результаты усвоения таблицы детьми, обучавшимися только на собственных ошибках, на ошибках и случайных примерах, а также на основе нейросетевой модели.

Результаты обучения учащихся и их сравнение с различными нейросетевыми парадигмами позволят среди многих моделей нейронных сетей выбрать те, которые наиболее похожи на работу мозга, что очень важно для совершенствования методов искусственного интеллекта. Можно указать, что группа под руководством проф. Д. Андерсона на основе идей запоминания математических фактов создала ряд интересных моделей, нашедших применение в областях обработки сенсорной информации. Один их классиков теории нейронных сетей Хехт-Нильсен в предложенной им теории работы коры головного мозга опирается на схожие модели.

Список литературы.

1. J. A. Anderson, The BSB Model: A simple nonlinear autoassociative neural network. In M. Hassoun (ed.) Associative Neural Memories. New York: Oxford U. Press. 1993.

2. J. A. Anderson, Seven times seven is about fifty. In S. Sternberg (Ed.) Invitation to Cognitive Science, Volume 4, Cambridge, MA: MIT Press. 1995.

3. J. A. Anderson, Arithmetic on a Parallel Computer: Perception Versus Logic, Brain and Mind, 4, 169-188, 2003.

4. J. A. Anderson, A Brain-Like Computer for Cognitive Software Applications: The Ersatz Brain Project, IEEE International Conference on Cognitive Informatics, Irvine CA, 2005.

5. B. Edelman, H. Abdi, and D. Valentin Multiplication Number Facts: Modeling Human Performance With Connectionist Networks, Psychologica Belgica, Vol.36.

6. R. Dallaway Dynamics of arithmetic: A connectionist view of arithmetic skills, Cognitive Science Research Papers 306, Brighton, UK: University of Sussex, 1994.

7. M. McCloskey and A. M. Lindemann MATHNET: preliminary results from a distributed model of arithmetic fact retrieval, In J.I.D. Campbell (Ed.), The Nature and Origin of Mathematical Skills, 365-410, Amsterdam, NL: North Holland, 1992.

8. G. B. Christianson, S. Becker A Model for Associative Multiplication. NIPS. pp.17-23,1998.

9. R. Hecht-Nielsen, A theory of cerebral cortex. Institute for Neural Computation, University of California, San Diego, Tec. Rep. #0404. 2004. Available: http://inc2.ucsd.edu.

10. A. Tatuzov, Neural network models for teaching multiplication table in primary school. Accepted to IEEE WCCI 2006, Vancouver, Canada, July, 16-21, 2006

Рекуррентно-автоматные характеристики динамических систем

Твердохлебов В. А.,
главный научный сотрудник Института проблем точной механики и управления РАН, доктор технических наук, профессор, 410208, г.Саратов, ул. Рабочая, д. 24, , тел. (8452) 27489, e-mail: tverdokhlebov@newmail.ru

В законах функционирования дискретных динамических систем представлены связи входных сигналов и выходных сигналов в зависимости от состояний систем. Эти законы выражаются в двух формах в числовой и в символьной (без числовой интерпретации знаков). Последняя форма используется, как правило, в теории автоматов, что фактически исключает применение мощных математических идеализаций: актуальной бесконечности, понятие бесконечно малой величины, предельного перехода, суммирование бесконечных рядов и др. В статье рассматривается новый набор характеристик, которые можно использовать для выражения свойств законов функционирования дискретных детерминированных динамических систем как при символьной, так и при числовой формах законов функционирования.

Пусть $A=(S,X,Y,\delta,\lambda)$ – детерминированный автомат с конечными множествами входных сигналов X и выходных сигналов Y, где функция переходов $\delta:S\times X\to S$ и функция выходов $\lambda:S\times X\to Y$ имеют символьную форму. В основе новой формы определение функций δ и λ геометрическим образом γ_s положено геометрическое представление, в котором автоматное отображение $\rho_s:\{s\}\times X^*\to Y$, дополненное вводимым линейным порядком ω_1 на X^*, преобразуется в упорядоченное множество $(X^*\times Y, \omega_1)$. Линейные порядки ω_1 на X^* и ω_2 на Y определяются правилами:

Правило 1. На множестве X вводим некоторый линейный порядок ω_1 (который будем обозначать $\prec_1$)

Правило 2. Порядок ω_1 на X распространим до линейного порядка на множестве X^*, полагая, что

- для любых слов $p_1,p_2\in X^*$ неодинаковой длины ($|p_1|\neq|p_2|$) $|p_1|<|p_2|\to p_1\prec_1 p_2$;
- для любых слов $p_1, p_2\in X^*$, для которых $|p_1|=|p_2|$ и $p_1\neq p_2$, их отношение по порядку ω_1 повторяет отношение ближайших слева несовпадающих букв слов p_1 и p_2. (Линейный порядок ω_2 на множестве выходных сигналов Y определяется произвольно).

При таком определении порядков ω_1 и ω_2 автоматное отображение ρ_s в прямоугольной системе координат с осями абсцисс (X^*,ω_1) и ординат (Y, ω_2) представляется ломаной линией (см. рис.1) и простым критерием автоматности ломаных линий: ломаная линия является геометрическим образом функций δ и λ инициального автомата (A,s) тогда и только тогда, когда она определена на всем множестве X^*. Линейный порядок ω_1 на X^*, порождающий порядок на множестве $(X^*\times Y, \omega_1)$, позволяет представить функции δ и λ только последовательностью вторых координат $y_{i_1}, y_{i_2},\ldots, y_{i_t},\ldots$ точек геометрического образа γ_s. Такое сведение задания геометрического образа инициального автомата к заданию последовательности является основой для использования рекуррентных форм как средств

определения функций δ и λ. Пусть $u = y_{i_1}, y_{i_2}, \ldots, y_{i_t}, \ldots$ последовательность вторых координат точек геометрического образа γ_s и $F(z_1,z_2,\ldots,z_m)=z_{m+1}$ – рекуррентная форма порядка m для последовательности u, то есть, для любого $k \geq m$ для последовательности u выполняется равенство

$$y_{j_{k+1}} = F\left(y_{j_{k-m+1}}, y_{j_{k-m+2}}, \ldots y_{j_k}\right)$$

(Рекуррентная форма $F(z_1,z_2,\ldots,z_m)=z_{m+1}$ может рассматриваться как интерполяционная функция частного типа и использоваться для приближенного задания законов поведения объекта диагностирования).

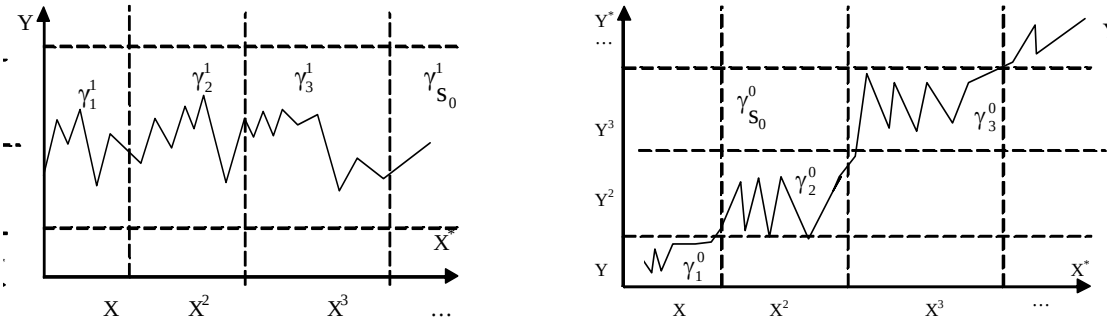


Рис.1. Геометрические образы автомата в двух вариантах систем координат:

I – системе координат с осями (X^*,ω_1), (Y, ω_2) и II - системе координат с осями (X^*,ω_1), (Y^*,ω_2).

Указанное представление функций δ и λ инициального автомата геометрическим образом с последующей заменой ломаной линии последовательностью вторых координат вершин ломаной позволяет классифицировать функции δ и λ по свойствам рекуррентных форм. Проведенные исследования показали, что рекуррентные формы определяют структуры в виде специфических ориентированных графов. Ориентация таких графов задает множество возможных вариантов обхода подграфов графа. Каждому обходу каждого подграфа соответствует последовательность, определяемая рекуррентной формой по правилам:

- множество символов последовательности однозначно отображается на множество вершин графа,
- в любом пути в графе, имеющем длину $m+1$ (где m – порядок рекуррентной формы), последовательность меток вершин определяется рекуррентной формой $y_{j_{k+1}} = F\left(y_{j_{k-m+1}}, y_{j_{k-m+2}}, \ldots y_{j_k}\right)$.

Справедливы следующие утверждения:

1) Каждая рекуррентная форма (имеющая указанную выше связь с графом) соответствует конечному числу конечных изолированных связных ориентированных графов.
2) Каждый такой граф имеет точно один цикл или одну петлю.
3) Вершины цикла или петля могут быть корнями деревьев.
4) Каждая конкретная последовательность, определяемая рекуррентной формой, совмещается точно с одним графом, в цикл или петлю которого имеется не более одного пути.

В геометрическом образе γ_s представлены целиком законы функционирования автомата, то есть все автоматное отображение. Конкретное функционирование автомата определяется сечением геометрического образа по вершинам, соответствующим последовательности увеличивающихся префиксов входной последовательность (см. рис.2). Задание инициальных автоматов геометрическими образами не сводится к определению структуры в форме ломаных линий. Фактически достаточно большой класс геометрических фигур может быть проинтерпретирован как геометрический образ автомата. Например, фигура, изображенная на рис.3. после ориентации ее обхода и выбора учитываемых точек на фигуре, представляется последовательностью $a_1,a_2,\ldots,a_d$. Эта последовательность может рассматриваться как задающая период периодического геометрического образа автомата.

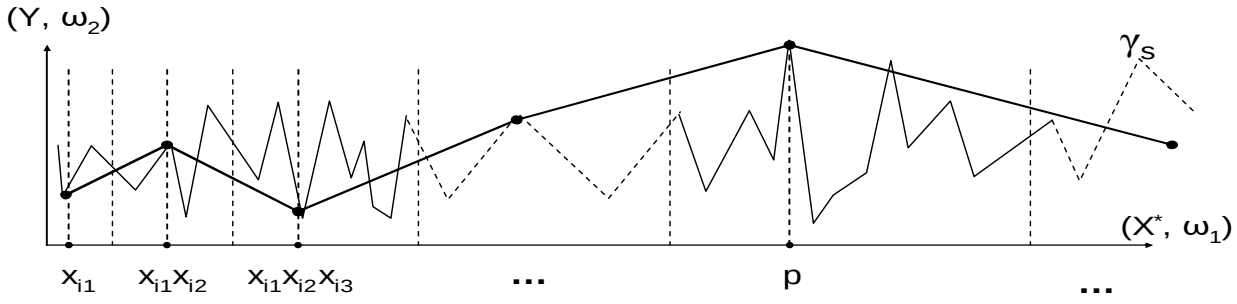
Для рекуррентно-автоматной характеристики законов функционирования конечных детерминированных автоматов предлагается использовать представление последовательностей вторых координат точек геометрического образа γ_s представлять в зависимости от размеров начального отрезка γ_s следующими параметрами:

- связью последовательности длин префиксов начального отрезка с номерами элементов последовательности, на которых требуется увеличивать порядок рекуррентной формы;

- определением наименьшего порядка рекуррентной формы, определяющей весь заданный начальный отрезок геометрического образа;

- связью для каждого $m = 1,2,\ldots$ числа значений дополнительного в рекуррентную форму $F(z_1,z_2,\ldots,z_m)=z_{m+1}$ параметра α, $H(z_1,z_2,\ldots,z_m\alpha)=z_{m+1}$, необходимого и достаточного для определения всего начального отрезка.

Рис.2. Геометрический образ конкретного функционирования автомата как сечение геометрического образа γ_s по точкам,



первые координаты которых являются префиксами, прикладываемой входной последовательности.

Полученные параметры образуют спектр характеристик функций δ и λ и, следовательно, инициальных автоматов. Такой спектр представляется в форме следующих таблиц:

Таблица 1

Длина префикса для γ_s	Наименьший порядок рекуррентной формы для префикса
1	m_1
2	m_2
3	m_3
…	…
k	m_k

Таблица 2

Порядок рекуррентной формы	Значение дополнительного параметра α
1	n_1
2	n_2
3	n_3
…	…
k-1	n_{k-1}

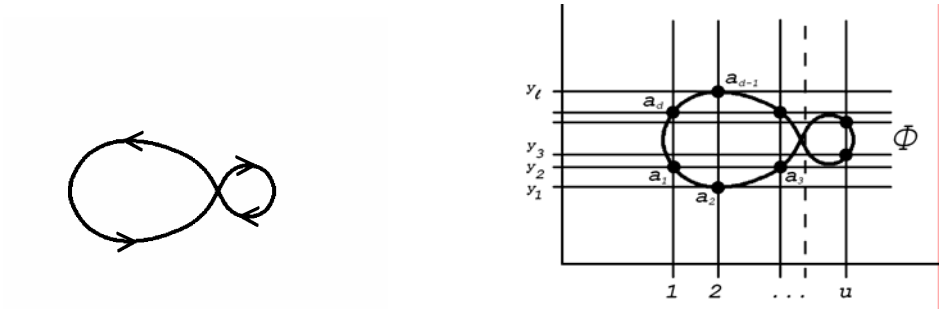


Рис.3. Преобразование обхода геометрической фигуры в геометрический образ функций δ и λ инициального автомата, где номера знаков в последовательности $a_1,a_2,\ldots,a_d$ соответствуют номерам входных слов в множестве (X^*,ω_1).

Рекуррентная форма для описания последовательности вторых координат точек геометрического образа γ_s совмещается с автоматной моделью. Для этого в состояниями автомата полагаются наборы вида $(z_1,z_2,\ldots,z_m\alpha)$, а входными сигналами элементы рассматриваемой последовательности, начиная с $m+1$ элемента. Состояние $\left(y_{j_{k-m+1}}, y_{j_{k-m+2}},\ldots y_{j_k}, \alpha\right)$ для входного сигнала $y_{j_{k+1}}$ изменяется на состояние $\left(y_{j_{k-m+2}},\ldots y_{j_k}, y_{j_{k+1}}, \beta\right)$, где $\alpha=\beta$, если $y_{j_{k-m+1}}, y_{j_{k-m+2}},\ldots y_{j_k}, y_{j_{k+1}}$ входит в последовательность. Такая рекуррентно-автоматная форма определяет последовательность, если формой представима вся последовательность. На основании этого рекуррентно-автоматная форма характеризуется порядком рекуррентной формы и числом значений параметра α (числом состояний автомата).

Список литературы

1. Твердохлебов В. А. Распознавание автоматов на основе геометрической интерпретации / Тезисы докладов XI Международной конференции «Проблемы теоретической кибернетики», Москва, 1996, с. 191

2. Твердохлебов В. А. Геометрические модели и методы в техническом диагностировании / Ж-л «Информационно - управляющие системы на железно дорожном транспорте» №3,4, Украина, Харьков, 1996, с.58.

3. Твердохлебов В. А. Синтез и анализ геометрических образов конечных автоматов / Тезисы докладов XII Международной конференции «Проблемы теоретической кибернетики», часть II, Москва, 1999, с.225.

4. Твердохлебов В. А. Рекуррентность геометрических образов / Научно техн. ж-л «Информационно-управляющие системы на железнодорожном транспорте», вып. 4-5, 2004, Харьков с. 88-90.

5. Твердохлебов В. А. Геометрические образы конечных детерминированных автоматов / Известия саратовского университета (Новая серия) том 5, вып.1, 2005, с. 141-153.

6. Твердохлебов В. А. Техническое диагностирование изменений параметров и свойств систем // Радіоелектронні і комп'ютерні системи, 2006, №6. С.119-123.

7. Твердохлебов В. А. Геометрические образы поведения дискретных детерминированных систем // Радіоелектронні і комп'ютерні системи, 2006, №5. С. 161-165.

Алгоритмы с оценками для дискретной задачи сегментации

Тебуева Фариза Билялovна, *к.ф.-м.н., доцент*
Шенкао Тимур Мухамедович,
Карачаево-Черкесская государственная технологическая академия,
кафедра прикладной математики,
369001, г. Черкесск, ул. Ставропольская, 36,
служ. телефон (87822) 33103, e-mail: timshenkao@yandex.ru

В настоящей работе в качестве адекватной математической модели задачи сегментации рынка [1] предлагается теоретико-графовая модель. Ее математическое описание формируется в предположении, что априори определена база для сегментации рынка, т.е. задано множество потенциальных покупателей (как индивидуальных потребителей, так и организаций) и выбраны факторы (критерии) сегментации. Кроме того, определена номенклатура однотипного товара, предъявляемого рынку. В процессе моделирования сегментации рынка формулируется математическая постановка многокритериальной задачи на двудольном графе $G = (V_1, V_2, E)$ [2], мощности долей которого $|V_1| = m$ и $|V_2| = l$, $m \le l$ [3].

Содержательно вершины $v_i \in V_1$ ($v_j \in V_2$) поставлены во взаимнооднозначное соответствие предъявленным типам товара $i = 1,2,...,m$ (группам потребителей $j = 1,2,...,l$), n_j – прогнозируемое количество покупаемых единиц товара представителями j-ой группы. Ребро $e = (v_i, v_j)$ принадлежит множеству E тогда и только тогда, когда i-й тип товара может оказаться приемлемым для покупателей j-ой группы ($1 \le j \le l$). Каждое ребро $e \in E$ графа взвешено числами $w_\upsilon(e)$, $\upsilon = \overline{1, N}$, где веса $w_\upsilon(e)$ отражают собой экспертно определенную степень потребительской пригодности i-го типа товара для покупателей из группы j, $0 \le w_\upsilon(e) \le 1$, $\upsilon = 1,2,...,N$, $e \in E$ (индексом υ перенумерованы критерии потребительского качества товара: долговечность, надежность, удобство в эксплуатации и т.д.). Через k_i обозначаем априорно заданное минимально допустимое количество экземпляров товара i-ого типа, при котором его производство оказывается экономически выгодным, $i = 1,2,...,m$.

Допустимым решением задачи сегментации на двудольном графе $G=(V_1,V_2,E)$ является такой его подграф $x=\left(V_1^x,V_2,E_x\right)$, $V_1^x\subseteq V_1$, $E_x\subseteq E$, каждая компонента связности которого представляет собой $(h+1)$- вершинную звезду [1], $h\in\{1,2,3,...,l\}$, центром которой является некоторая вершина $v_i\in V_1$ и ребра которой образуют множество E_x^i, $i\in\{1,2,...,m\}$. При этом висячие вершины конкретной звезды E_x^i образуют подмножество $V_2^x(v_i)\subseteq V_2$, удовлетворяющее неравенству

$$\sum_{v_j\in V_2^i} n_j\geq k_i,\ v_i\in V_1^x, \tag{1}$$

где центр $v_i\in V_1^x$, $i=1,2,...,m$ и объединение $\bigcup_{v_i\in V_1^x} V_2^x(v_i)=V_2$. $X=X(G)=\{x\}$ – множество всех допустимых решений (МДР). На МДР X определена векторная целевая функция (ВЦФ)

$$F(x)=\left(F_1(x),F_2(x),...,F_{N+1}(x)\right), \tag{2}$$

состоящая из N критериев весового вида $MAXSUM$

$$F_\upsilon(x)=\sum_{e\in E_x} w_\upsilon(x)\to\max,\ \upsilon=\overline{1,N} \tag{3}$$

и одного критерия комбинаторного вида

$$F_{N+1}(x)=\left|V_1^x\right|\to\max,\ \upsilon=\overline{1,N}, \tag{4}$$

отражающего разнообразие номенклатуры, то есть количество типов (товара), которые целесообразно производить.

ВЦФ (2)-(3) определяет собой в МДР X паретовское множество (ПМ) $\tilde{X}$ [3], состоящее из всех паретовских оптимумов (ПО) $x\in X$ [3]. Всякая пара ПО $\tilde{x}_1,\tilde{x}_2\in\tilde{X}$ считается эквивалентной, если выполняется равенство значений ВЦФ: $F(\tilde{x}_1)=F(\tilde{x}_2)$. Поэтому в настоящей работе рассматриваем алгоритмическую проблему нахождения так называемого полного множества альтернатив (ПМА) [3]. Подмножество $X^0\in\tilde{X}$ называется ПМА, если его мощность $\left|X^0\right|$ минимальна при выполнении равенства $F(X^0)=F(\tilde{X})$, где $F(X^*)=\{F(x):x\in X^*\}$, $\forall X^*\subseteq X^0$.

В реальных условиях значения весов $w_\upsilon(e)$, $\upsilon=\overline{1,N}$, задаваемые экспертами, имеют приближенный характер. Для отражения неопределенности подобного рода в математической модели возможно использование аппарата интервального исчисления [4]: вес ребра $e\in E$ представляем в виде интервала $w(e)=\left[w^1(e),w^2(e)\right]$, $w^1(e)\leq w^2(e)$. В интервальной постановке вместо ВЦФ (2)–(4) рассматриваем ВЦФ

$$F'(x)=\left\{W(x),\left|V_1^x\right|\right\}, \tag{5}$$

которая состоит из интервального критерия вида *MAXSUM*

$$W(x)=\sum_{e\in E_x} w(e)\to\max, \tag{6}$$

и одного критерия комбинаторного вида (аналогичного (4))

$$\left|V_1^x\right|\to\max. \tag{7}$$

В работе [5] обосновывается утверждение о том, что всякая интервальная задача на графах с интервальной целевой функцией (ИЦФ) (6) эквивалентна соответствующей производной двукритериальной задаче с ВЦФ $F'(x)=\left(F^1(x),F^2(x)\right)$, $F^\upsilon(x)=\sum_{e\in E_x} w^\upsilon(x)\to\max$, $\upsilon=\overline{1,2}$. На основании этого методы решения многокритериальных задач можно использовать для интервальных задач. Предлагается следующий двухуровневый подход к решению сформулированной многокритериальной задачи. На нижнем уровне осуществляется перебор различных комбинаций множеств V_1^x, удовлетворяющих условию (1). Для этого различные сочетания из m вершин множества V_1 по $k\in\{1,2,...,m\}$ вершин перенумеруем в порядке неубывания их мощностей индексом $r=\overline{1,M}$, $M=2^m-1$; r-ое сочетание представляем в виде подмножества $V^r\subseteq V_1$, совокупность этих

подмножеств обозначаем через $W_1 = \{V^r\}$, $r = \overline{1, M}$. Тогда через $X_r \subset X$ обозначим такое подмножество допустимых решений $x = (V_1^x, V_2, E_x)$, у каждого из которых имеет место совпадение $V_1^x = V^r$. Отметим, что для некоторых подмножеств V^r подмножество X_r может оказаться пустым.

На верхнем уровне для фиксированного множества центров звезд $V^r \subseteq V_1$ вначале выбирается конечное подмножество $\Lambda_N^0 \subset \Lambda_N$ множества $\Lambda_N = \left\{ \lambda = (\lambda_1, \lambda_2, ..., \lambda_N) : \sum_{\nu=1}^{N} \lambda_\nu = 1, \lambda_\nu > 0, \nu = \overline{1, N} \right\}$. Далее для каждого вектора $\lambda = (\lambda_1, \lambda_2, ..., \lambda_N) \in \Lambda_N^0$ строится линейная свертка критериев (JICK)

$$F^{(\lambda)}(x) = \sum_{\upsilon=1}^{N} \lambda_\nu F_\nu(x), \; x \in X^r, \; 1 \le r \le M. \tag{8}$$

В силу аддитивной природы критериев (3) ЛСК (8) представляем в качестве целевой функции (ЦФ)

$$F^{(\lambda)}(x) = \sum_{\nu=1}^{N} \lambda_\nu \sum_{e \in E_x} w_\nu(e) \to \max, \tag{9}$$

где для допустимого решения $x = (V_1^x, V_2, E_x) \in X^r$ множество ребер E_x определяется таким допустимым разбиением доли V_2 на подмножества

$$V_2^x(v_i) \subseteq V_2, \; v_i \in V^r, \; \bigcup_{v_i \in V^r} V_2^x(v_i) = V_2, \tag{10}$$

которое обеспечивает выполнение неравенств

$$\sum_{v_j \in V_2^x(v_i)} n_j \ge k_i, \; v_i \in V^r. \tag{11}$$

При фиксированном множестве центров $V_1^x = V^r$ формирование подмножеств $V_2^x(v_i)$, $v_i \in V_1^x$ которые определяют допустимое решение x, удовлетворяющее условию (11), получается в процессе достижения требуемого экстремального значения ЦФ (9).

Как известно [6], если при фиксированном векторе $\lambda \in \Lambda_N$ допустимое решение представляет собой оптимум по ЛСК (8), то это решение является парето-оптимальным.

Для данного графа $G = (V_1, V_2, E)$ в результате объединения найденных множеств решений $\tilde{X}(V^r)$ по всевозможным вариантам $V^r \in W_1$ множеств центров звезд, получаем некоторое подмножество паретовского множества $\tilde{X} = \tilde{X}(G)$. В свою очередь из полученного подмножества ПМ $\tilde{X}$ выделяется некоторое подмножество искомого ПМА $X^0 = X^0(G)$. Это подмножество можно рассматривать в качестве аппроксимации ПМА X^0.

Обозначим через α приближенный алгоритм градиентного типа для нахождения оптимального покрытия данного двудольного графа $G = (V_1, V_2, E)$ при фиксированном множестве центров $V^r \subseteq V_1$ для случая, когда в условии (8) значения его параметров удовлетворяют равенствам $n_j = n$ для каждого $j = \overline{1, l}$ и значение $k_i = nl / m_r$ для каждого $i \in V^r$. Эти условия означают, что все допустимые решения $x = (V_1^x, V_2, E_x) \in X^r$ представляют собой покрытия данного графа G звездами одинаковой степени. Отсюда, не теряя общности, рассматриваемую задачу при указанных условиях можно рассмотреть в следующей постановке.

Для данного одновзвешенного двудольного графа $G = (V_1, V_2, E)$, с мощностями его долей $|V_1| = m$ и $|V_l| = l$, l кратно m (то есть $l = mn$, где $n = l / m$), МДР представляет собой такой остовной подграф $x = (V_1, V_2, E_x)$ графа G, у которого каждая компонента связности представляет собой $(n+1)$-вершинную звезду с центром в некоторой вершине $v_i \in V_1$.

Предлагаемый алгоритм α нахождения покрытия оптимального по значению ЦФ $F(x)=\sum_{e\in E_x} w(e)\to\max$ состоит из подготовительного этапа и n вычислительных этапов. На подготовительном этапе множество вершин второй доли V_2 разбиваем на n равномощных подмножеств V_2^s, $\left|V_2^s\right|=m$, $s=\overline{1,n}$, $n=\frac{l}{m}$. Далее для каждого подмножества $V_2^s\subset V_2$ строим двудольный граф $G^s=\left(V_1,V_2^s,E^s\right)$ такой, что его множество E^s состоит из ребер $e=(v_i,v_j)\in E$, у каждого из которых концевая вершина $v_i\in V_1$ и концевая вершина $v_j\in V_2^s$. Результатом подготовительного этапа является последовательность двудольных графов $G^s=\left(V_1,V_2^s,E^s\right)$, $s=\overline{1,n}$.

Последующие n этапов перенумеруем индексом $s=\overline{1,n}$. Вычислительная работа очередного этапа s состоит в нахождении в графе G совершенного паросочетания $E_0^s\subseteq E^s$, оптимального по значению целевой функции

$$W(E_x^s)=\sum_{e\in E_x^s} w(e)\to\max .$$

Последний этап $s=n$ завершается построением остовного подграфа $x_\alpha=(V_1,V_2,E_{x_\alpha})$, множество ребер которого $E_{x_\alpha}=\bigcup_{s=1}^{n}E_0^s$. По определению алгоритма α остовной подграф является допустимым решением задачи покрытия данного графа G звездами одинаковой степени.

Обоснование оценок эффективности алгоритма α осуществляется в терминах теории вероятностных графов. Будем использовать следующие обозначения: $\varphi=\varphi(m)$ – сколь угодно медленно растущая функция от m, $\lim_{m\to\infty}\varphi(m)=\infty$; $J\left(m,l,R,N\right)=\left\{G\right\}$ – множество N -взвешенных двудольных графов $G=(V_1,V_2,E)$, с мощностями долей $\left|V_1\right|=m$, $\left|V_2\right|=l$, $l=mn$, в каждом из которых всякому ребру $e\in E$ приписаны веса $w_\nu(e)\in\left\{1,2,\ldots,R\right\}$, $\nu=\overline{1,N}$. В данной работе получен следующий результат.

Теорема 1. *Если* $R\le\frac{m}{2\ln m+\varphi}$ *и* $m=O(l)$, *то для почти всех графов* $G\in J(m,l,R,N)$ *алгоритм* α *находит ПМА задачи сегментации (задачи покрытия графа звездами одинаковой степени) с ВЦФ (2)-(4), причем ПМА является одноэлементным* $\left|X^0\right|=1$ *и сложность его нахождения* $\tau(\alpha)\le O(m^{3/2}l)$.

Список литературы

1. Макдоналд М., Данбар Я. Сегментирование рынка: практическое руководство. – М.: Дело и сервис, 2002.

2. Лекции по теории графов /Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. – М.: Наука, 1990.

3. Емеличев В. А., Перепелица В. А. Сложность дискретных многокритериальных задач// Дискретная математика. – 1994. – Вып. 1,6. – С. 3-33.

4. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. – М.: Мир, 1987.

5. Perepelitsa V. A. and Kozina G. L. Interval Discrete Models and Multiobjectivity. Complexity Estimates // Interval Computations. 1993. 1.

6. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. – М.: Наука, 1982.

Распознавание графов с отмеченными вершинами конечным автоматом

Тихончев М. Ю.,
доцент Филиала Ульяновского государственного университета в г. Димитровграде,
433512, Россия, Ульяновская обл., г.Димитровград, пр.Ленина, 29-15,
E-mail: dns@niiar.ru

Одной из основных моделей в теории управляющих систем и вычислительных процессов является модель взаимодействия управляющего автомата и операционной среды (управляемой системы). Это взаимодействие часто представляется как процесс перемещения автомата по лабиринту. Это привело к возникновению области исследований поведения автоматов в лабиринтах [1,2]. В качестве лабиринтов рассматриваются различные графы с отмеченными ребрами или вершинами.

В настоящей работе рассматривается задача распознавания конечным автоматом графов с отмеченными вершинами, известных как детерминированные графы [3,4]. Полагаем, что автомат может оставлять в вершинах графа специальные дополнительные нестираемые отметки (не стирая при этом их основных отметок) и, находясь в любой вершине, воспринимать основные и дополнительные отметки этой вершины и всех вершин, ей смежных. Такие автоматы известны как автоматы с красками [5].

Пусть $\mathbf{M}$ - заданный алфавит отметок. $G=(\mathrm{G},\mathrm{E}_G,\mathrm{X}_G,\mu)$ - конечный, простой (т.е. без петель и кратных рёбер), связный, неориентированный граф с отмеченными вершинами, у которого G - конечное множество вершин, $|\mathrm{G}|\geq 2$, E_G - множество рёбер, $\mathrm{X}_G \subseteq \mathbf{M}$ - множество отметок вершин, μ: $\mathrm{G}\to\mathrm{X}$ - сюръективная функция разметки вершин. Ребро графа, соединяющее вершины s и t, будем обозначать <st>. Через deg(s) будем обозначать степень вершины s.

Будем говорить, что графы G и H изоморфны, и обозначать это $G\cong H$, если существует биекция φ: $\mathrm{G}\to\mathrm{H}$ такая, что $<\mathrm{g}_i\mathrm{g}_j>\in\mathrm{E}_G$ тогда и только тогда, когда $<\varphi(\mathrm{g}_i)\varphi(\mathrm{g}_j)>\in\mathrm{E}_H$, и $\mu(\mathrm{g})=\mu(\varphi(\mathrm{g}))$ для всех $\mathrm{g}\in\mathrm{G}$.

Через $K(\mathbf{M})$ обозначим класс всех попарно неизоморфных графов, определенных на алфавите $\mathbf{M}$. В соответствие с работами [3,4] граф G будем называть детерминированным, если для любых $\mathrm{g,s,t}\in\mathrm{G}$ из $\{<\mathrm{gs}>,<\mathrm{gt}>\}\subseteq\mathrm{E}_G$ и $\mathrm{s}\neq\mathrm{t}$ следует $\mu(\mathrm{s})\neq\mu(\mathrm{t})$. Через $K_d(\mathbf{M})$ обозначим подкласс всех детерминированных графов из класса $K(\mathbf{M})$. Далее для простоты рассуждений будем полагать, что $\mathbf{M}=\{1,2,...,\mathrm{m}\}$, $\mathrm{m}=|\mathbf{M}|$.

Под обходом графа $G\in K(\mathbf{M})$ будем понимать его обход методом поиска в глубину с расстановкой дополнительных отметок. При этом полагаем, что дополнительные отметки суть натуральные числа, вершине начала обхода приписывается дополнительная отметка 1, всем остальным вершинам при первом посещении приписывается дополнительная отметка (i+1), где i - дополнительная отметка предшествующей при обходе вершины. Также считаем, что изначально всем вершинам из G приписана "пустая" дополнительная отметка 0.

Пусть $\mathbf{M}_{add}(G)=\{0,1,...,\mathrm{k}\}$, где k - максимальная из дополнительных отметок, приписываемых вершинам графа G при его обходе. Нетрудно видеть, что $\mathrm{k}\leq|\mathrm{G}|$. Известно, что при обходе графа G поиском в глубину требуется совершить $2|\mathrm{G}|-2$ тактов перемещения.

Через $\mathbf{a}_i=(\mathrm{a}_{i1},...,\mathrm{a}_{ip})$, $\mathrm{a}_{ij}\in\mathbf{M}\times\mathbf{M}_{add}(G)$, $\mathrm{p}\leq|\mathrm{G}|$, $\mathrm{i}=0,...,2|\mathrm{G}|-2$, $\mathrm{j}=1,...,\mathrm{p}$, обозначим вектор, элементами a_{ij} которого являются пары (α_{ij},β_{ij}) основных и дополнительных отметок текущей вершины и всех смежных ей вершин на i-ом такте обхода графа G ($\mathbf{a}_0$ соответствует началу обхода). Причем, (α_{i1},β_{i1}) - соответственно основная и дополнительная отметки текущей вершины. Остальные элементы вектора $\mathbf{a}_i$ упорядочены так, что $\alpha_{ij}\leq\alpha_{ij+1}$ и, если $\alpha_{ij}=\alpha_{ij+1}$, то $\beta_{ij}\leq\beta_{ij+1}$, $\mathrm{j}=2,...,\mathrm{p}-1$. Упорядоченное мультимножество векторов $\mathbf{X}(G)=\{\mathbf{a}_0,...,\mathbf{a}_{2|G|-2}\}$ назовем лабиринтной характеристикой графа G. В общем случае граф G может иметь несколько различных лабиринтных характеристик. Множество всех различных лабиринтных характеристик графа G назовем его полной лабиринтной характеристикой и обозначим $\mathbf{XF}(G)=\{\mathbf{X}_1(G),...,\mathbf{X}_r(G)\}$. Как следует из определения обхода, r не превосходит $|\mathrm{G}|!$.

Следующая теорема устанавливает связь между изоморфизмом графов из класса $K(\mathbf{M})$ и совпадением их лабиринтных характеристик.

Теорема. *Для любых графов $G,H\in K(\mathbf{M})$ следующие утверждения эквивалентны:*
1) $G\cong H$;
2) $\mathbf{XF}(G)=\mathbf{XF}(H)$;
3) $\mathbf{XF}(G)\cap\mathbf{XF}(H)\neq\emptyset$.

Согласно этой теореме задача проверки изоморфизма графов G и H эквивалентна проверке

условия $\mathbf{XF}(G)\cap\mathbf{XF}(H)\neq\emptyset$. Однако, поскольку $\mathbf{XF}(G)$ и $\mathbf{XF}(H)$ могут содержать до |G|! и |H|! различных лабиринтных характеристик соответственно, такая проверка может оказаться чрезмерно сложной. Покажем, что для случая $G\in K_d(\mathbf{M})$ проверка условия $\mathbf{XF}(G)\cap\mathbf{XF}(H)\neq\emptyset$ может быть осуществлена конечным автоматом, число состояний и время функционирования которого полиномиально зависят от |G|.

Пусть G - произвольный граф из $K_d(\mathbf{M})$. Требуется построить автомат $A(G)$, способный, перемещаясь по произвольному графу $H\in K(\mathbf{M})$, за конечное число тактов определить является H изоморфным G или нет. В начальный момент автомат устанавливается в произвольную вершину графа H. Будем говорить, что автомат $A(G)$ распознаёт граф G в класса $K(\mathbf{M})$.

Под конечным автоматом $A(G)$ понимаем конечный, всюду определенный, инициальный автомат Мили [6] с заключительными состояниями, т.е. семёрку $<\mathrm{A,B,S},\varphi,\psi,\mathrm{S}_e,\mathrm{s}_0>$, где A и B - входной и выходной алфавиты, соответственно, S - конечное множество состояний, φ и ψ - функции переходов и выходов, соответственно, $\mathrm{S}_e\subset\mathrm{S}$ - множество заключительных состояний, $\mathrm{s}_0\in\mathrm{S}\backslash\mathrm{S}_e$ - начальное состояние. Перейдя в любое из заключительных состояний, автомат останавливается.

Входной алфавит A представляет из себя множество всевозможных векторов $\mathbf{a}=(\mathrm{a}_1,...,\mathrm{a}_{p+1})$ длины (p+1), где $\mathrm{p}=\max\limits_{g\in G}\deg(\mathrm{g})$ (нетрудно видеть, что $\mathrm{p}\leq|\mathrm{X}_G|$). Первые l ($1\leq l\leq\mathrm{p}+1$) элементов вектора $\mathbf{a}$ являются парами $\mathrm{a}_i=(\alpha_i,\beta_i)$, $1\leq\mathrm{i}\leq l$, из $\mathbf{M}\times\mathbf{M}_{add}(G)$. Причем пары $(\alpha_2,\beta_2),...,(\alpha_l,\beta_l)$ упорядочены так, что $\alpha_j\leq\alpha_{j+1}$ и, если $\alpha_j=\alpha_{j+1}$, то $\beta_j\leq\beta_{j+1}$, j=2,...,l−1. Остальные элементы суть пустые символы Λ. Выходной алфавит B - множество пар $(\mathrm{b}_1,\mathrm{b}_2)$, где $\mathrm{b}_1\in(\mathbf{M}_{add}(G)\backslash\{0\})\cup\{\Lambda\}$, $\mathrm{b}_2\in\mathbf{M}\cup\{\Lambda\}$.

Вход автомата интерпретируется следующим образом. Пусть в некоторый момент времени $A(G)$ находится в вершине h графа H и deg(h)=d. Возможны два случая:

1. d≤p. Тогда автомат получает на вход вектор $\mathbf{a}=(\mathrm{a}_1,...,\mathrm{a}_{p+1})$, где $\mathrm{a}_i=(\alpha_i,\beta_i)$, $\alpha_i\in\mathbf{M}$, $\beta_i\in\mathbf{M}_{add}(G)$, для i=1,...,d+1, и $\mathrm{a}_i=\Lambda$ для i>d+1. $\mathrm{a}_1=(\alpha_1,\beta_1)$ - пара основной и дополнительной отметок вершины h, соответственно. $\mathrm{a}_2,...,\mathrm{a}_{d+1}$ - пары основных и дополнительных отметок всех вершин, смежных h. Остальные элементы вектора $\mathbf{a}$ (если они есть) суть символы Λ.

2. d>p. В этом случае $A(G)$ получает на вход вектор $\mathbf{a}$, первый элемент которого как и в первом случае есть пара значений основной и дополнительной отметок текущей вершины, остальные p элементов - символы Λ.

Выход автомата интерпретируется следующим образом. Пусть $\psi(\mathbf{a},\mathrm{s})=(\mathrm{b}_1,\mathrm{b}_2)$. Если $\mathrm{b}_1\neq\Lambda$, то автомат отмечает текущую вершину дополнительной отметкой b_1. При $\mathrm{b}_1=\Lambda$ текущей вершине не приписывается никакой новой дополнительной отметки. Если $\mathrm{b}_2\neq\Lambda$, то $A(G)$ перемещается в вершину t, $<\mathrm{ht}>\in\mathrm{E}_H$, такую, что $\mu(\mathrm{t})=\mathrm{b}_2$. При $\mathrm{b}_2=\Lambda$ автомат остается в текущей вершине. Заметим, что $\mathrm{b}_2\neq\Lambda$ всегда однозначно определяет вершину для перемещения, если исследуемый граф является детерминированным.

Определим так трактуемую функцию выходов. Пусть $\psi(\mathbf{a},\mathrm{s})=\psi(\mathbf{a})$ зависит только от входного вектора $\mathbf{a}=(\mathrm{a}_1,...,\mathrm{a}_{+1})$ и не зависит от текущего состояния автомата. Если $\mathrm{a}_2=\Lambda$, то $\psi(\mathbf{a})=(\Lambda,\Lambda)$. В противном случае $\psi(\mathbf{a})$ определяется по следующим правилам.

Если $\beta_1=0$, то $\mathrm{b}_1=\max\limits_{a_i\neq\Lambda}\beta_i+1$, иначе $\mathrm{b}_1=\Lambda$.

Если для i≥2 найдется $\mathrm{a}_i=(\alpha_i,0)$ и $\beta_j\neq0$ для всех j=2,...,i−1, то $\mathrm{b}_2=\alpha_i$.

Пусть $\mathrm{a}_i\neq(\alpha_i,0)$ для всех i≥2. Тогда

если $\beta_1=0$, то $\mathrm{b}_2=\alpha_j$, где j выбирается из условия $\beta_j>\beta_k$ для всех k≠j;

если $\beta_1=1$, то $\mathrm{b}_2=\Lambda$;

если $\beta_1>1$, то $\mathrm{b}_2=\beta_1-1$.

Определим множество состояний S. Положим $\mathrm{S}=\{\mathrm{s}_0,\mathrm{s}_1^*,\mathrm{s}_2^*\}\cup\{\mathrm{s}_1^i,...,\mathrm{s}_{2|G|-2}^i|\mathrm{i}=1,...,|\mathrm{G}|\}$. Состояние s_0 объявим начальным. Положим $\mathrm{S}_e=\{\mathrm{s}_1^*,\mathrm{s}_2^*\}$.

Перед заданием функции переходов проведем следующее построение. Для каждой вершины g_i графа G, $1\leq\mathrm{i}\leq|\mathrm{G}|$ построим характеристику $\mathbf{X}_i(G)=\mathbf{a}_0^i,...,\mathbf{a}_{2|G|-2}^i$, соответствующую обходу графа, начинающемуся из вершины g_i. Причем при построении $\mathbf{X}_i(G)$ наложим на правила обхода графа дополнительное требование. Если при обходе графа среди вершин, смежных текущей, имеется более чем одна вершина с дополнительной отметкой 0, то перемещение осуществляется в ту из них, чья основная отметка минимальна. В силу детеминированности графа G, это требование обеспечивает однозначность построения каждой из характеристик $\mathbf{X}_i(G)$.

Функцию переходов $\varphi(\mathbf{a},\mathrm{s})$ определим следующим образом.

$$\varphi(\mathbf{a},\mathrm{s}_0)=\begin{cases} s_1^i, & \text{если } \mathbf{a}=\mathbf{a}_0^i \\ s_2^*, & \text{если } \mathbf{a}\neq\mathbf{a}_0^i \end{cases},\ 1\leq\mathrm{i}\leq|\mathrm{G}|;$$

$$\varphi(\mathbf{a},\mathrm{s}_j^i)=\begin{cases} s_{j+1}^i, & \text{если } \mathbf{a}=\mathbf{a}_j^i \\ s_2^*, & \text{если } \mathbf{a}\neq\mathbf{a}_j^i \end{cases},\ 1\leq\mathrm{j}\leq2|\mathrm{G}|-3,\ 1\leq\mathrm{i}\leq|\mathrm{G}|;$$

$$\varphi(\mathbf{a},s^i_{2|G|-2})=\begin{cases} s^*_1, \text{ если } \mathbf{a}=\mathbf{a}^i_{2|G|-2} \\ s^*_2, \text{ если } \mathbf{a}\neq\mathbf{a}^i_{2|G|-2} \end{cases}, 1\leq i\leq |G|;$$

Полученный автомат является, в общем случае, недетерминированным. Применим к его множеству состояний и функции переходов следующую процедуру.

Если при некотором значении входного символа $\mathbf{a}$ и некотором k, $1\leq k\leq 2|G|-2$, функция φ допускает переходы из некоторого состояния s в несколько различных состояний $s^{i1}_k,\ldots,s^{ij}_k$, то

1) отождествим все эти состояния $s^{i1}_k,\ldots,s^{ip}_k$ в одно состояние s';

2) если при некотором $\mathbf{a}=\mathbf{a}^*$ функция φ допускает значение $\varphi(\mathbf{a}^*,s')=s\neq s^*_2$, изменим φ, убрав из нее возможность перехода из s' в s^*_2 по входному символу $\mathbf{a}^*$.

Будем повторять описанную процедуру до тех пор, пока автомат не станет детерминированным. Требуемый автомат $A(G)$ построен.

Автомат $A(G)$ пытается обойти граф H методом поиска в глубину, начав обход из его произвольной вершины. Если автомат переходит в заключительное состояние s^*_1, то для некоторого i, $1\leq i\leq |G|$, $\mathbf{X}_i(G)\subset\mathbf{XF}(H)$ и, согласно теореме, $G\cong H$. Переход автомата в состояние s^*_2 означает, что $\{\mathbf{X}_1(G),\ldots,\mathbf{X}_{|G|}(G)\}\cap\mathbf{XF}(H)=\emptyset$ и, следовательно, $G\ncong H$. Время функционирования автомата $A(G)$ не превосходит $2|G|-1$ тактов и мощность множества его состояний, очевидно, не превосходит $|G|(2|G|-2)+3$.

Список литературы

1. Килибарда Г., Кудрявцев В. Б., Ушчумлич Щ. Независимые системы автоматов в лабиринтах // Дискретная математика, 2003, Т. 15, вып. 2, С. 3 – 39.

2. Килибарда Г., Кудрявцев В. Б., Ушчумлич Щ. Коллективы автоматов в лабиринтах // Дискретная математика, 2003, Т. 15, вып. 3, С. 3 – 40.

3. Сапунов С. В. Эквивалентность помеченных графов // Труды ИПММ НАНУ, 2002, –Т. 8, – С. 162–167.

4. Грунский И. С., Курганский А. Н. Языки графов с помеченными вершинам // Труды ИПММ НАНУ, 2004, –Т. 9, – С. 53–60.

5. Насыров А. З. Об обходе лабиринтов автоматами, оставляющими нестираемые метки. Дискретная математика. 1997, Т. 9, № 1, С. 123–133

6. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение теорию автоматов. –М.: Наука, 1985, – 320 с.

О сложности кратных диагностических экспериментов для подмножеств состояний автомата

Уваров Д. В.,

механико-математический факультет МГУ им. М. В. Ломоносова

E-mail: dmitri.uvarov@gmail.com

Введение

Важным классом экспериментов с автоматами являются диагностические эксперименты. Такие эксперименты используются для идентификации неизвестного состояния автомата. В общем случае, диагностические эксперименты рассматриваются для некоторого класса K инициальных автоматов. Эксперимент называется *диагностическим*, если результаты его применения к любым двум различным автоматам из K различны.

В качестве класса K мы будем рассматривать множество некоторых инициальных автоматов V_q, получающихся при различном выборе начального состояния у заданного автомата V. Таким образом, задача сводится к тому, чтобы отличать пары состояний у одного автомата. Э.Ф. Мур в своей работе [2] доказал, что для того чтобы отличить одну пару состояний в автомате приведенного вида, в худшем случае может потребоваться слово длины $n-1$, где n — количество состояний автомата, причем существует автомат на котором эта оценка достигается. Здесь будет рассмотрено некоторое обобщение этой задачи, когда вместо одной пары состояний рассматриваются целые множества, и отличать надо любые пары из этих множеств, с помощью кратных безусловных диагностических экспериментов. Были получены точные оценки сложности таких экспериментов. В

качестве сложности эксперимента мы будем рассматривать его объем, то есть сумму длин слов, из которых состоит эксперимент.

Определения и формулировка результатов

Пусть $V = (A, Q, B, \varphi, \psi)$ — автомат приведенного вида, $|Q| = n \geq 2$. Рассмотрим подмножество $Q' \subset Q$, $|Q'| = l \geq 2$. Поскольку V — автомат приведенного вида, то для Q' существует некоторый *диагностический эксперимент* $E_{V,Q'}$, то есть такое множество слов из A^*, что любые два состояния из Q' отличимы каким-нибудь из этих слов.

Пусть $v(E)$, где E — эксперимент, обозначает объем эксперимента, т.е. сумму длин входящих в него слов. Нас интересует, какой объем в наихудшем случае может иметь эксперимент $E_{V,Q'}$. Введем следующую функцию Шеннона:

$$L(n,l) = \max_{V:|Q|=n,\, Q':|Q'|=l} \left(\min_{E_{V,Q'}} v(E_{V,Q'}) \right).$$

Здесь минимум берется по всем диагностическим экспериментам $E_{V,Q'}$, а максимум — по всем автоматам приведенного вида с n состояниями и подмножествам с l состояниями.

Получен следующий результат.

Теорема 1 $L(n,l) = (l-1)(n - \frac{l}{2})$.

Расширим понятие диагностического эксперимента, приняв в рассмотрение выделенные попарно не пересекающиеся множества

$$Q_1, Q_2, \ldots, Q_m \subset Q, \quad |Q_i| = l_i \geq 1, i = 1, \ldots, m.$$

Диагностическим экспериментом для подмножеств $Q_1, Q_2, \ldots, Q_m$ множества состояний автомата V назовем такое множество слов $E'_{V,Q_1,\ldots Q_m}$, что любые два состояния, лежащие в различных подмножествах отличимы некоторым словом из $E'_{V,Q_1,\ldots Q_m}$. *Внутренним диагностическим экспериментом для подмножеств* $Q_1, Q_2, \ldots, Q_m$ множества состояний автомата V назовем такое множество слов $E''_{V,Q_1,\ldots Q_m}$, что любые два различных состояния, лежащие в одном подмножестве отличимы некоторым словом из $E''_{V,Q_1,\ldots Q_m}$. Заметим, что если подмножества $Q_1, Q_2, \ldots, Q_m$ одноэлементные, то диагностический эксперимент для них является диагностическим экспериментом для множества $Q' = Q_1 \cup \ldots \cup Q_m$. Также, если у нас имеется всего одно подмножество, то внутренний диагностический эксперимент совпадает с обычным диагностическим экспериментом для него. Как и выше, введем следующие функции Шеннона:

$$L'(n, l_1, \ldots, l_m) = \max_{V:|Q|=n,\, Q_1:|Q_1|=l_1,\ldots,Q_m:|Q_m|=l_m} \left(\min_{E'_{V,Q_1,\ldots Q_m}} v(E'_{V,Q_1,\ldots Q_m}) \right),$$

$$L''(n, l_1, \ldots, l_m) = \max_{V:|Q|=n,\, Q_1:|Q_1|=l_1,\ldots,Q_m:|Q_m|=l_m} \left(\min_{E''_{V,Q_1,\ldots Q_m}} v(E''_{V,Q_1,\ldots Q_m}) \right).$$

Были получены точные значения для этих функций:

Теорема 2 $L'(n, l_1, \ldots, l_m) = (l-1)(n - \frac{l}{2})$, *где* $l = l_1 + \ldots + l_m$.

Теорема 3 $L''(n, l_1, \ldots, l_m) = (l-m)(n - \frac{l-m+1}{2})$, *где* $l = l_1 + \ldots + l_m$.

Рассмотрим еще одну задачу. Пусть $Q'' \subset Q' \subset Q$, $|Q'| = l'$, $|Q''| = l''$. Пусть для множества Q'' уже построен некоторый диагностический эксперимент $E_{V,Q''}$. Мы хотим расширить этот эксперимент до диагностического для множества Q', то есть найти такой диагностический эксперимент $E_{V,Q'}$, что $E_{V,Q''} \subset E_{V,Q'}$. Введем следующую функцию Шеннона, которая описывает, на какой объем в наихудшем случае нужно расширить эксперимент $E_{V,Q''}$.

$$\widehat{L}(n, l', l'') = \max_{V:|V|=n,\, Q'':|Q''|=l'',\, Q':|Q'|=l',\, E_{V,Q''}} \left(\min_{E_{V,Q'} \supset E_{V,Q''}} (v(E_{V,Q'}) - v(E_{V,Q''})) \right)$$

Для этой функции также была найдена точная оценка:

Теорема 4 $\widehat{L}(n,l',l'') = (l'-l'')(n-\dfrac{l'-l''+1}{2})$.

Еще одна задача возникает, если ввести ограничение на длину слова, отличающего какие-либо два состояния в автомате. Пусть $\mathcal{V}_n^d$ — множество автоматов с n состояниями такие, что любую пару их состояний можно отличить словом длины не большей, чем d. Аналогично определим функцию Шеннона:

$$L(n,l,d) = \max_{V\in\mathcal{V}_n^d,\, Q':|Q'|=l} \left(\min_{E_{V,Q'}} v(E_{V,Q'}) \right).$$

Для этой функции верно следующее.

Теорема 5

$$L(n,l,d) = \begin{cases} d(n-d) + \dfrac{1}{2}(d+n-l)(d-n+l-1), & \textit{если } d \geq n-l-1; \\ d(l-1), & \textit{иначе.} \end{cases}$$

В заключение автор выражает благодарность А.С. Подколзину за научное руководство и В.Б. Кудрявцеву за внимание к работе.

Список литературы

1. Уваров Д. В. О сложности кратных диагностических экспериментов для подмножеств состояний автомата // Интеллектуальные системы. Том 9, стр. 485-503.

2. Мур Э. Ф. Умозрительные эксперименты с последовательностными машинами. - В кн. Автоматы. - М.: ИЛ, 1956, с. 179-210.

3. Кудрявцев В. Б., Подколзин А. С., Ушчумлич Ш. Введение в теорию абстрактных автоматов: Учеб. пособие / МГУ. М., 1985.

4. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.

Случай произвольной частоты запросов в задаче поиска по маске

Уварова Т. Д.,

кафедра математической теории интеллектуальных систем,
механико-математический факультет МГУ им. М. В. Ломоносова
E-mail: tatiana.uvarova@gmail.com

Введение

В работе исследуется следующая задача информационного поиска, называемая задачей интервального поиска на булевом кубе. Имеется некоторое подмножество V n-мерного булева куба B_2^n, называемое библиотекой. На булевом кубе берется произвольный интервал (u,w), где $u=(u_1,\ldots,u_n)$, $w=(w_1,\ldots,w_n)$ и $u \preceq w$, то есть $u_i \leq w_i$ $i=1,\ldots,n$. Требуется определить все такие элементы $y \in V$, $y=(y_1,\ldots,y_n)$, называемые записями, для которых выполнено $u \preceq y \preceq w$.

Задача интервального поиска на булевом кубе соответствует поиску по маске. Те компоненты с номером i, для которых $u_i = w_i$, задают маску, и в этих компонентах значения у искомых объектов должны совпадать с маской, то есть $y_i = u_i = w_i$. А для тех i, для которых $0 = u_i \neq w_i = 1$, компонента y_i может принимать произвольное значение из $\{0,1\}$.

Задачу можно решать, если на каждом шаге алгоритма проверять условие $u_j \leq y_j \leq w_j$, $j \in \{1,\ldots,n\}$ для одной компоненты записи, причем номер компоненты одинаков для всех записей на одном шаге алгоритма.

Классу таких алгоритмов удобно сопоставить сбалансированные информационные деревья.

Для средней (по всем запросам) временной сложности сбалансированных деревьев показано следующее. Существуют положительные числа $\alpha < \beta$, такие что если k — мощность библиотеки V, то сложность дерева, решающего ЗИП с библиотекой V ограничена снизу величиной k^α и сверху — величиной k^β.

Если вероятностная мера, заданная на множестве запросов такова, что запрос с i неизвестными компонентами появляется с вероятностью в a раз меньшей, чем запрос с $i-1$ неизвестной компонентой, то сложность такой задачи равна $O(k^{\log_2(1+1/(1+2a))})$. В частности, при $a=1$, мы получаем случай равномерной вероятностной меры, рассмотренный в [2].

В заключение автор выражает благодарность Э.Э.Гасанову за внимание и помощь в работе.

Основные понятия и формулировка результатов

Мы будем использовать терминологию и обозначения из работы [1], но поскольку в данной работе рассматриваются только древовидные схемы, то здесь будет приведена несколько упрощенная версия понятия информационного графа.

Если X — множество символов запросов с заданным на нем вероятностным пространством $\langle X, \sigma, \mathbf{P}\rangle$, где σ — алгебра подмножеств множества X, $\mathbf{P}$ — вероятностная мера на σ; Y — множество символов данных (записей); ρ — бинарное отношение на $X \times Y$, называемое отношением поиска; то пятерка $S = \langle X, Y, \rho, \sigma, \mathbf{P}\rangle$ называется *типом*. Тройка $I = \langle X, V, \rho\rangle$, где V — некоторое конечное подмножество множества Y, называемое библиотекой, называется задачей информационного поиска (ЗИП) типа S. Содержательно ЗИП $I = \langle X, V, \rho\rangle$ состоит в перечислении для произвольно взятого запроса $x \in X$ всех и точно тех записей $y \in V$, что $x\rho y$. Если $\mathcal{F}$ — суть множества символов одноместных предикатов, определенных на X, $\mathcal{F}$ называется базовым множеством и описывает множество элементарных операций, используемых при решении задачи информационного поиска.

Над базовым множеством $\mathcal{F}$ определяется понятие *информационного графа* (ИГ). В конечной многополюсной ориентированной сети выбирается вершина — полюс, называемая корнем. Остальные полюса называются листьями и им приписываются записи из Y. Ребрам ИГ приписываются предикаты из множества $\mathcal{F}$. Таким образом нагруженную многополюсную ориентированную сеть называют информационным графом над базовым множеством $\mathcal{F}$. Затем определяется *функционирование ИГ*. Предикатное ребро проводит запрос $x \in X$, если предикат ребра истинен на x; ориентированная цепочка ребер проводит x, если каждое ребро цепочки проводит x; запрос x проходит в вершину β ИГ, если существует ориентированная цепь, ведущая из корня в вершину β, которая проводит x; запись y, приписанная листу α, попадает в ответ ИГ на x, если x проходит в лист α. Ответом ИГ U на запрос x называют множество записей, попавших в ответ U на x, и обозначают его $\mathcal{J}_U(x)$. Эту функцию $\mathcal{J}_U(x)$ считают результатом функционирования ИГ U.

ИГ U *решает* ЗИП $I = \langle X, V, \rho\rangle$, если $\mathcal{J}_U(x) = \{y \in V : x\rho y\}$.

Вводится *сложность ИГ*. Предикат $\varphi_\beta(x)$ истинный на x, если x проходит в вершину β, и ложный в противном случае, называется функцией фильтра вершины β. *Сложностью ИГ U на запросе $x \in X$* называется число $T(U,x) = \sum_{\beta\in\mathcal{R}} \psi_\beta \cdot \varphi_\beta(x)$, где $\mathcal{R}$ — множество вершин ИГ U, ψ_β — количество ребер, исходящих из вершины β. Эта величина равна числу функций, вычисленных алгоритмом поиска, определяемым ИГ U, на запросе x.

Если каждая функция из $\mathcal{F}$ — измерима (относительно алгебры σ), то для любого ИГ U над $\mathcal{F}$ функция $T(U,x)$ измерима.

Сложностью ИГ U называется математическое ожидание величины $T(U,x)$, равное $T(U) = \mathbf{M}_x T(U,x)$. Она характеризует среднее время поиска. Если f предикат на множестве X, то $N_f(x) = \{x \in X : f(x) = 1\}$.

Легко показать, что

$$T(U) = \sum_{\beta\in U} \psi_\beta \cdot P(N_{\varphi_\beta}(x)). \tag{1}$$

Сложностью ребра, исходящего из вершины β назовем число $P(N_{\varphi_\beta}(x))$. Согласно [**?**] сложность ИГ равна сумме сложностей ребер.

Рассмотрим следующую ЗИП. Имеется некоторое k-элементное подмножество n-мерного булева куба $V \in B_2^n$ (библиотека). На булевом кубе задан некоторый интервал (u,w), где $u = (u_1,\ldots,u_n)$, $w = (w_1,\ldots,w_n)$, и $u \preceq w$, то есть $u_i \le w_i$, $\forall i = 1,\ldots,n$. Требуется определить все элементы $y \in V$, удовлетворяющие условию $u \preceq y \preceq w$.

Очевидно, что если $u_i = 1$ для некоторого i, то и $w_i = 1$, а, следовательно, и $y_i = 1$. Аналогично, если $w_i = 0$, то и $y_i = 0$. Таким образом, вышеописанная ЗИП сводится к следующей: есть библиотека $V \in B_2^n$, $|V| = k$, берем запрос $x = (x_1,\ldots,x_n)$– трехзначный вектор, компоненты которого могут быть равны либо 1, либо 0, либо 2: если $u_i = 1$, то $x_i = 1$, если $w_i = 0$, то $x_i = 0$, иначе $x_i = 2$. Для данного запроса $x = (x_1,\ldots,x_n)$ хотим найти все $y = (y_1,\ldots,y_n) \in V$, для которых $y_i = x_i$, если $x_i = 1$ или $x_i = 0$, и y_i любое из $\{0,1\}$, если $x_i = 2$.

Получаем тип задач $S_n =< B_3^n, B_2^n, \rho, \sigma, P >$, где $B_3^n = \{0,1,2\}^n$ и $B_2^n = \{0,1\}^n$ трехзначный и двузначный (булев) кубы, соответственно, ρ : $x\rho y \Leftrightarrow (x_i = y_i) \vee (x_i = 2)$, σ — множество подмножеств B_3^n, и P – произвольная вероятностная мера на B_3^n.

Множество задач $I =< B_3^n, V, \rho >$ типа S_n, где $|V| = k$, вероятностная мера равна P, обозначим через $\mathcal{I}(n, k, P)$.

Пусть $x \in \{0, 1, 2\}$ и $y \in \{0, 1, 2\}$. Определим функцию x^y :

$$x^y = \begin{cases} x, & \text{если} \quad (y = 1)\&(x \neq 2) \\ \bar{x}, & \text{если} \quad (y = 0)\&(x \neq 2) \\ 1, & \text{если} \quad (y = 2) \vee (x = 2) \end{cases},$$

причем $\bar{x}$ понимается здесь как булевое отрицание.

Вершину со степенью полуисхода, равной нулю назовем *висячей вершиной. Информационным деревом (ИД)* назовем ИГ без циклов, множество листьев которого совпадает с множеством висячих вершин, и все ребра которого ориентированы от корня к листьям. *Высотой ИД* назовем длину максимального пути из корня в лист. Будем говорить, что вершина v находится на ярусе с номером $i = 0, \ldots, n$, если длина пути из корня в вершину v равна i.

Будем рассматривать базис переменных $\mathcal{F} = \{x_i^{s_i} | s_i \in \{0, 1\}, i = 1, \ldots, n\}$. Пусть $\sigma = (i_1, \ldots, i_n)$ — перестановка длины n. Сбалансированным деревом с характеристикой h, соответствующим перестановке σ, назовем следующее дерево над базисом $\mathcal{F}$. Для всех $p = 0, \ldots, h-1$ из каждой вершины яруса с номером p исходит точно два ребра, нагрузка одного из них равна $x^0_{i_{p+1}}$, нагрузка второго — $x^1_{i_p}$. Из некоторых вершин яруса с номером h исходят непересекающиеся цепочки ребер длины $n-h$, суммарная нагрузка такой цепочки содержит переменные $x_{i_{h+1}}, \ldots, x_{i_n}$, нагрузки двух различных цепочек различны.

Обозначим $B_3^n(i, a) = \{x = (x_1, \ldots, x_n) \in B_3^n | x_i = a\}$, $P_a^i = P(B_3^n(i, a))$.

Теорема 1 *Пусть $I \in \mathcal{I}(n, k, P)$. И пусть вероятностное пространство, заданное на B_3^n таково, что $|\{i : P_a^i = 1, a \in \{0, 1, 2\}\}| = 0$. Тогда при $n \to \infty$, $k \to \infty$ существуют числа $0 < \alpha < \beta < 1$ такие, что*

$$k^\alpha \preccurlyeq T(I) \preccurlyeq k^\beta.$$

Для любого $x \in B_3^n$ обозначим через $\|x\|$ число компонент, равных 2. Пусть на B_3^n задана вероятностная функция $P(x) = p_a(x) = b/a^{\|x\|}$. Фиксируем параметр a.

Теорема 2 *При $n \to \infty$, $k \to \infty$ для любой ЗИП $I \in \mathcal{I}(n, k, p_a)$ выполнено*

$$T(I) \asymp k^{\log_2(1+1/(2a+1))}.$$

Следствие 1 *Для любого числа $0 < \alpha < 1$ существует такая вероятностная функция $p(x)$, заданная на B_3^n, что для любой ЗИП $I \in \mathcal{I}(n, k, p)$ выполнено*

$$T(I) \asymp k^\alpha.$$

Оценки сложности для произвольной вероятностной меры

Вспомогательные утверждения

Обозначим

$$\begin{aligned} A &= \max\{(P_0^i + 2P_2^i + P_1^i) : i \in \{1, \ldots, n\}\} \\ B &= \max\{(P_j^i + P_2^i) : j \in \{0, 1\}, i \in \{1, \ldots, n\}\} \\ a &= \min\{(P_0^i + 2P_2^i + P_1^i) : i \in \{1, \ldots, n\}\} \\ b &= \min\{(P_j^i + P_2^i) : j \in \{0, 1\}, i \in \{1, \ldots, n\}\} \end{aligned}$$

Лемма 1 *Пусть в вершину v из корня ведет цепь длины m с суммарной нагрузкой длины m. Тогда для любого ребра e, выходящего из вершины v верно*

$$b^m \leq T(e) \leq B^m.$$

Доказательство. Пусть цепочке ребер, ведущей из корня в вершину v приписана суммарная нагрузка $x_{i_1}^{s_1} \& \ldots \& x_{i_m}^{s_m}$. Заметим, что $\varphi_v(x) = 0$, если существует $j \in \{1, \ldots, m\}$, для которого $x_{i_j} = \overline{s_j}$, иначе $\varphi_v(x) = 1$. Следовательно,

$$T(e) = M_x T(e, x) = P(\varphi_v(x)) = P(\bigwedge_{j=1}^{m} (x_{i_j} = s_j) \vee (x_{i_j} = 2)) = \prod_{j=1}^{m} (P_{s_j}^{i_j} + P_2^{i_j}),$$

откуда и следует утверждение леммы.

Обозначим через $E_m(D)$ ярус ребер в информационном дереве D, исходящих из вершин яруса с номером $(m-1)$. Будем называть их ребрами яруса с номером m. Если E — множество ребер ИД, то через $T(E)$ будем обозначать сумму сложностей ребер из множества E.

Лемма 2 *Пусть в дереве D к ребру e из корня ведет цепь ребер с нагрузками $x_{i_1}^{\sigma_1}, x_{i_2}^{\sigma_1}, \ldots, x_{i_m}^{\sigma_m}$, а в дереве D' к ребру e' из корня ведет цепь ребер с нагрузками $x_{i_2}^{\sigma_1}, \ldots, x_{i_m}^{\sigma_m}$. Тогда*

$$T(e) = T(e') \cdot (P_{\sigma_1}^{i_1} + P_2^{i_1}).$$

Доказательство. Пусть ребро e исходит из вершины β. По определению

$$\begin{aligned} T(e) &= P(N_{\varphi_\beta}(x)) = \\ &= P(\bigwedge_{j=1}^{m}(x_{i_j} = \sigma_j) \vee (x_{i_j} = 2)) = \\ &= \prod_{j=1}^{m}(P_{\sigma_j}^{i_j} + P_2^{i_j}) = T(e') \cdot (P_{\sigma_1}^{i_1} + P_2^{i_1}). \end{aligned}$$

Лемма 3 *Для сбалансированного дерева D с характеристикой h и $1 \le m \le h$ верно*

$$2a^{m-1} \le T(E_m(D)) \le 2A^{m-1}.$$

Доказательство. Индукция по номеру яруса m.

База индукции. E_1 представляет собой два ребра, исходящих из корня, для которых в любом случае вычисляется значение приписанной им нагрузки. То есть $T(E_1(D)) = 2$.

Индуктивный переход. Пусть для произвольного дерева D' с характеристикой $h' \ge m-1 \ge 1$ выполнено

$$2a^{m-2} \le T(E_{m-1}(D')) \le 2A^{m-2}.$$

Дерево D состоит из двух ребер, исходящих из корня, из концевой вершины каждого из которых, в свою очередь, исходят сбалансированные поддеревья D_1 и D_2 с характеристикой $h-1$. Ребру, ведущему к поддереву D_1, приписана функция x_{i_1}, ребру, ведущему к поддереву D_2 — $\overline{x}_{i_1}$. Очевидно, что $E_m(D) = E_{m-1}(D_1) \cup E_{m-1}(D_2)$, и каждому ребру e из $E_m(D)$ соответствует некоторое единственное ребро e' из $E_{m-1}(D_j)$ ($j = 1$ или $j = 0$). Из утверждения леммы 2 следует, что $T(e) = T(e') \cdot (P_{\sigma_1}^{i_1} + P_2^{i_1})$. Заметим, что $T(E_{m-1}(D_1)) = T(E_{m-1}(D_2))$, и значит

$$\begin{aligned} T(E_m(D)) &= \sum_{e \in E_m(D)} T(e) = \\ &= \sum_{e' \in E_{m-1}(D_1)} (P_1^{i_1} + P_2^{i_1})T(e') + \sum_{e' \in E_{m-1}(D_2)} (P_0^{i_1} + P_2^{i_1})T(e') = \\ &= (P_1^{i_1} + P_2^{i_1}) \cdot T(E_{m-1}(D_1)) + (P_0^{i_1} + P_2^{i_1}) \cdot T(E_{m-1}(D_2)) = \\ &= T(E_{m-1}(D_1)) \cdot (P_1^{i_1} + 2P_2^{i_1} + P_0^{i_1}). \end{aligned}$$

Воспользовавшись предположением индукции, получаем

$$2a \cdot a^{m-2} \le T(E_m(D)) \le 2A \cdot A^{m-2},$$

$$2a^{m-1} \le T(E_m(D)) \le 2A^{m-1}.$$

Обозначим через $D_h(k, \sigma)$ сбалансированное дерево с характеристикой h, соответствующее перестановке σ и имеющее k концевых цепочек.

Лемма 4

$$T(D_h(k, \sigma)) - T(D_{h-1}(k, \sigma)) \ge 2a^{h-1} - kB^{h-1}$$

Доказательство. Деревья $D_h(k, \sigma)$ и $D_{h-1}(k, \sigma)$ отличаются только ярусом ребер с номером h. А именно, в дереве $D_h(k, \sigma)$ ярус с номером h находится в сбалансированной части, и для его сложности верна оценка $T(E_h(D_h(k, \sigma))) \ge 2a^{h-1}$. В дереве $D_{h-1}(k, \sigma)$ ярус $E_h(D_{h-1}(k, \sigma)))$ состоит из k ребер, сложность каждого из которых, как было доказано в лемме 1, меньше, чем B^{h-1}. Отсюда и следует утверждение леммы.

Лемма 5 $T(D_h(\sigma)) - T(D_{h+1}(\sigma)) \ge -2A^h + kb^h$

Доказательство. Деревья $D_h(k,\sigma)$ и $D_{h+1}(k,\sigma)$ отличаются только ярусом ребер с номером $h+1$. А именно, в дереве $D_{h+1}(k,\sigma)$ ярус с номером $h+1$ находится в сбалансированной части, и для его сложности верна оценка $T(E_{h+1}(D_{h+1}(k,\sigma))) \le 2A^h$. В дереве $D_h(k,\sigma)$ ярус $E_{h+1}(D_h(k,\sigma)))$ состоит из k ребер, сложность каждого из которых, как было доказано в лемме 1, больше, чем b^h. Отсюда и следует утверждение леммы.

Лемма 6 *Для характеристики h оптимального сбалансированного дерева, решающего ЗИП $I \in \mathcal{I}(n,k,P)$ верно*

$$h \ge \log_{A/b} \frac{k}{2}.$$

Доказательство. Пусть $D_h(k,\sigma)$ — оптимальное сбалансированное дерево, решающее ЗИП I. Предположим, что $h < \log_{A/b}(k/2)$. Тогда, в силу утверждения леммы 5,

$$\begin{aligned} T(D_h(k,\sigma)) - T(D_{h+1}(k,\sigma)) &\ge kb^h - 2A^h > \\ &> kb^{\log_{A/b}(k/2)} - 2A^{\log_{A/b}(k/2)} = \\ &= k\left(\frac{k}{2}\right)^{\log_{A/b} b} - 2\left(\frac{k}{2}\right)^{\log_{A/b} A} = \\ &= k^{\log_{A/b} A}\left(\frac{1}{2^{\log_{A/b} b}} - \frac{1}{2^{\log_{A/b} A - 1}}\right) = 0, \end{aligned}$$

что противоречит оптимальности дерева $D_h(k,\sigma)$.

Лемма 7 *Для характеристики h оптимального сбалансированного дерева, решающего ЗИП $I \in \mathcal{I}(n,k,P)$ верно*

$$h \le \log_{a/B} \frac{k}{2} + 1.$$

Доказательство. Пусть $D_h(k,\sigma)$ — оптимальное сбалансированное дерево, решающее ЗИП I. Предположим, что $h \ge \log_{a/B}(k/2) + 1$. Тогда, в силу утверждения леммы 4,

$$\begin{aligned} T(D_h(k,\sigma)) - T(D_{h-1}(k,\sigma)) &\ge 2a^{h-1} - kB^{h-1} > \\ &> 2a^{\log_{a/B}(k/2)} - kB^{\log_{a/B}(k/2)} = \\ &= 2\left(\frac{k}{2}\right)^{\log_{a/B} a} - k\left(\frac{k}{2}\right)^{\log_{a/B} B} = \\ &= k^{\log_{a/B} a}\left(\frac{1}{2^{\log_{a/B} a - 1}} - \frac{1}{2^{\log_{a/B} B}}\right) = 0, \end{aligned}$$

что противоречит оптимальности дерева $D_h(k,\sigma)$.

Лемма 8 *Пусть дерево $D_h(k,\sigma)$ является оптимальным решающим деревом для ЗИП $I \in \mathcal{I}(n,k,P)$. Тогда*

$$T(D_h(k,\sigma)) > \frac{1}{a-1} \cdot k^{\log_{A/b} a} - \frac{2}{a-1}.$$

Доказательство. Воспользуемся результатами лемм 3, 6. Получим

$$\begin{aligned} T(D_h(k,\sigma)) &\ge \sum_{i=1}^{h} T(E_i(D_h(k,\sigma))) \ge \\ &\ge 2(1 + a + \cdots + a^{h-1}) = \\ &= 2\frac{a^h - 1}{a-1} \ge \frac{2}{a-1}\left(a^{\log_{A/b}(k/2)} - 1\right) = \\ &= \frac{2^{1-\log_{A/b} a}}{a-1} \cdot k^{\log_{A/b} a} - \frac{2}{a-1} > \\ &> \frac{1}{a-1} \cdot k^{\log_{A/b} a} - \frac{2}{a-1}. \end{aligned}$$

Лемма 9 *Пусть дерево $D_h(k,\sigma)$ — оптимальное сбалансированное дерево, решающее ЗИП $I \in \mathcal{I}(n,k,P)$. Тогда*

$$T(D_h(k,\sigma)) \le \left(\frac{2}{2B-1} + \frac{B}{1-B}\right) k^{\log_2 2B}.$$

Доказательство. Рассмотрим дерево $D_{[\log_2 k]}(k,\sigma)$, решающее ЗИП I. Получаем, что

$$\begin{aligned} T(D_h(k,\sigma)) &\leq T(D_{[\log_2 k]}(k,\sigma)) \leq \\ &\leq 2\sum_{i=1}^{[\log_2 k]}(2B)^{i-1} + B^{[\log_2 k]}k\sum_{i=1}^{n-[\log_2 k]}B^{i-1} \leq \\ &\leq \frac{2}{2B-1}(2B)^{[\log_2 k]} + B^{[\log_2 k]}\cdot k\cdot\frac{1}{1-B} \leq \\ &\leq \frac{2}{2B-1}k^{\log_2 2B} + \frac{B}{1-B}k^{\log_2 2B}. \end{aligned}$$

Доказательство теоремы 1

Покажем, что существуют числа $0 < \alpha \leq \beta < 1$, такие, что для $I \in \mathcal{I}(n,k,P)$ $k^\alpha \preccurlyeq T(I) \preccurlyeq k^\beta$ при $k \to \infty$, $n \to \infty$.

Пусть $D_h(k,\sigma)$ — оптимальное решающее дерево. Из леммы 8 следует, что

$$T(D_h(k,\sigma)) > \frac{1}{a-1}\cdot k^{\log_{A/b} a} - \frac{2}{a-1},$$

причем, поскольку $A \geq a > 1$, а $b < 1$, то $A/b > a > 1$ и $0 < \log_{A/b} a < 1$. Из утверждения леммы 9 следует, что

$$T(D_h(k,\sigma)) \leq \left(\frac{2}{2B-1} + \frac{B}{1-B}\right)k^{\log_2 2B}.$$

Положим $\alpha = \log_{A/b} a < \beta = \log_2 2B < 1$.

Теорема доказана.

Существование задач с заданной сложностью

Вспомогательные утверждения

Лемма 10 *Пусть на множестве B_3^n задана вероятностная мера p_a, для которой $p_a(x) = b/a^{\|x\|}$. Тогда числа a и b связаны следующим соотношением:*

$$b = \left(\frac{a}{2a+1}\right)^n.$$

Доказательство.

$$\begin{aligned} 1 &= \sum_{x\in B_3^n} p(x) = \sum_{i=0}^{n} C_n^i \cdot 2^{n-i}\frac{b}{a^i} = \\ &= b\cdot 2^n\sum_{i=1}^{n} C_n^i\left(\frac{1}{2a}\right)^i = b\cdot 2^n\left(1+\frac{1}{2a}\right)^n = b\left(\frac{2a+1}{a}\right)^n. \end{aligned}$$

Ребро произвольного дерева, решающего ЗИП $I \in \mathcal{I}(n,k,p_a)$, к которому ведет цепочка, суммарная нагрузка которой не содержит одинаковых переменных и (без ограничения общности) равная $f = x_1^{s_1}\&\cdots\&x_i^{s_i}$, назовем i-ребром.

Пусть $\|x\|_i$ — число двоек среди первых i компонент вектора $x \in B_3^n$. Обозначим $B^n(j) = \{x \in B_3^n | f(x) = 1, \|x\|_i = j\}$.

Лемма 11 *Для произвольного i-ребра e выполнено*

$$\sum_{x\in B^n(j)} p_a(x)T(e,x) = C_i^j\frac{1}{a^j}\left(\frac{a}{2a+1}\right)^i.$$

Доказательство. Обозначим $B_\alpha^n(j) = \{x \in B^n(j) | (x_1\cdots x_i) = \alpha\}$. Фиксируем вектор α. Очевидно, что $|B_\alpha^n(j)| = 3^{n-i}$ (вне зависимости от α) и $B^n(j) = \bigsqcup_{\alpha: f(\alpha)=1} B_\alpha^n(j)$. Кроме того, $|\{B_\alpha^n(j) | f(\alpha) = 1\}| = C_i^j$. Среди запросов из $B_\alpha^n(j)$ в точности $C_{n-i}^m\cdot 2^{n-i-m}$ запросов содержат $(j+m)$ двоек среди своих компонент — j двоек среди первых i компонент и m двоек среди последних $n-i$ компонент,

$m = 0, \ldots, n-i$. Следовательно,

$$\begin{aligned}
\sum_{x \in B^n(j)} p_a(x) T(e,x) &= \sum_{\alpha: f(\alpha)=1} \sum_{x \in B^n_\alpha(j)} \sum_{m=0}^{n-i} C^m_{n-i} \cdot 2^{n-i-m} \cdot \frac{1}{a^{j+m}} \left(\frac{a}{2a+1}\right)^n = \\
&= \frac{2^{n-i}}{a^j} \left(\frac{a}{2a+1}\right)^n \sum_{\alpha: f(\alpha)=1} \sum_{x \in B^n_\alpha(j)} \sum_{m=0}^{n-i} C^m_{n-i} \left(\frac{1}{2a}\right)^m = \\
&= \frac{2^{n-i}}{a^j} \left(\frac{a}{2a+1}\right)^n \sum_{\alpha: f(\alpha)=1} \sum_{x \in B^n_\alpha(j)} \left(\frac{2a+1}{2a}\right)^{n-i} = \\
&= \frac{2^{n-i}}{a^j} \left(\frac{a}{2a+1}\right)^n \left(\frac{2a+1}{2a}\right)^{n-i} \cdot |\{B^n_\alpha(j) | f(\alpha) = 1\}| = \\
&= C^j_i \cdot \frac{1}{a^j} \left(\frac{a}{2a+1}\right)^i.
\end{aligned}$$

Лемма 12 *Пусть e — произвольное i-ребро дерева, решающего ЗИП $I \in \mathcal{I}(n,k,p_a)$. Тогда*

$$T(e) = \left(\frac{a+1}{2a+1}\right)^i.$$

Доказательство.

$$\begin{aligned}
T(e) &= \sum_{x \in B^n_3} p_a(x) T(e,x) = \sum_{j=0}^{i} \sum_{x \in B^n(j)} p_a(x) T(e,x) = \\
&= \sum_{j=0}^{i} C^j_i \cdot \frac{1}{a^j} \left(\frac{a}{2a+1}\right)^i = \\
&= \left(\frac{a}{2a+1}\right)^i \cdot \left(\frac{a+1}{a}\right)^i = \left(\frac{a+1}{2a+1}\right)^i.
\end{aligned}$$

Обозначим $c = (a+1)/(2a+1)$.

Лемма 13 *Сложность $T(D)$ сбалансированого дерева D с характеристикой h, решающего ЗИП $I \in \mathcal{I}(n,k,p_a)$ равна*

$$T(D) = 2\frac{(2c)^h - 1}{2c-1} + k\frac{c^n}{c-1} - k\frac{c^h}{c-1}.$$

Доказательство.

$$\begin{aligned}
T(D) &= 2(1 + 2c + \cdots + (2c)^{h-1}) + c^h \cdot k(1 + c + \cdots + c^{n-h-1}) = \\
&= 2\frac{(2c)^h - 1}{2c-1} + c^h \cdot k\frac{c^h - 1}{c-1}.
\end{aligned}$$

Лемма 14 *Пусть D — оптимальное сбалансированное дерево, решающее ЗИП $I \in \mathcal{I}(n,k,p_a)$. Тогда его характеристика h равна $]\log_2 k[$.*

Доказательство. Предположим, что $h < \log_2 k - 1$. Рассмотрим дерево D' с характеристикой $h+1$, решающее ЗИП I. Тогда

$$T(D) - T(D') = -2(2c)^h + k \cdot c^h = c^h \cdot (-2 \cdot 2^h + k) > 0,$$

что противоречит оптимальности дерева D. Предположим, что $h > \log_2 k$. Рассмотрим дерево D'' с характеристикой $h-1$, решающее ЗИП I. Получим, что

$$T(D) - T(D'') = 2(2c)^{h-1} - k \cdot c^{h-1} = c^{h-1} \cdot (2^h - k) > 0,$$

что противоречит оптимальности дерева D.

Доказательство теоремы 2

Воспользуемся тем, что для характеристики h оптимального дерева D, решающего ЗИП I выполнено $h = [\log_2 k]$. Используя результат леммы 13 получаем

$$\begin{aligned} 2\frac{(2c)^{\log_2 k-1}-1}{2c-1} + k\frac{c^n - c^{\log_2 k}}{c-1} \leq \quad & T(I) \quad \leq 2\frac{(2c)^{\log_2 k}-1}{2c-1} + k\frac{c^n - c^{\log_2 k-1}}{c-1}, \\ \frac{2k^{\log_2 2c}}{2c(2c-1)} - \frac{k^{\log_2 2c}}{c-1} + \underline{O}(1) \leq \quad & T(I) \quad \leq \frac{2k^{\log_2 2c}}{2c-1} - \frac{k^{\log_2 2c}}{c(c-1)} + \underline{O}(1), \\ k^{\log_2 2c}\left(\frac{1}{c(2c-1)} - \frac{1}{c-1}\right) \preceq \quad & T(I) \quad \preceq k^{\log_2 2c}\left(\frac{2}{2c-1} - \frac{1}{c(c-1)}\right). \end{aligned}$$

Учитывая, что $c = (a+1)/(2a+1)$, получаем, что

$$T(I) \asymp k^{\log_2(1+1/(2a+1))}.$$

Доказательство следствия.
Фиксируем число $0 < \alpha < 1$. Возьмем параметр $a = (1-2^{\alpha-1})/(2^\alpha - 1)$. Тогда

$$2a+1 = \frac{2-2^\alpha+2^\alpha-1}{2^\alpha-1} = \frac{1}{2^\alpha-1}, \quad 1 + \frac{1}{2a+1} = 2^\alpha.$$

Из теоремы 2 следует, что для ЗИП $I \in \mathcal{I}(n,k,p_a)$ верно

$$T(I) \asymp k^{\log_2(1+1/(2a+1))} = k^\alpha,$$

что и доказывает следствие теоремы 2.

Список литературы

1. Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. *Москва, ФИЗМАТЛИТ*, 2002

2. Блайвас Т. Д. Асимптотика сложности задачи интервального поиска на булевом кубе в классе сбалансированных древовидных схем. *Дискретная математика*, том**16**, вып. 4, 2004 г., стр. 65-78

О сложности алгоритмов анализа и синтеза автоматов

Ш. Ушчумлич,
Технолого-металлургический факультет, Белградский университет,
Карнегиева, 4, 11001 Белград, Сербия
e-mail: suscum@gmail.com

В теоретических и прикладных аспектах автоматов важную роль играют так называемые задачи анализа и синтеза автоматов. Многообразие способов, описывающих функционирование автоматов, может быть разбито условно на две группы: группу языков, т.е., алгебро-логических описаний, и группу способов, описывающих внутренние процессы реальных автоматов, примером которых могут служить диаграммы Мура, источники и др. Переход от способа первого типа ко второму называется синтезом автоматов, а от второго к первому — анализом автоматов. В качестве способа первого типа в данной работе рассматривается язык регулярных выражений; в качестве способов второго типа — источники.

Среди конкретных алгоритмов анализа и синтеза автоматов фундаментальную роль играют алгоритмы анализа и синтеза автоматов, предложенные В.М. Глушковым [2]. Они были созданы одними из первых, имеют достаточно прозрачную и естественную структуру и повлекли за собой появление целого ряда модификаций. Важно отметить, что при решении задач анализа и синтеза автоматов внимание обращалось в первую очередь на принципиальную возможность их решения, а исследование их характеристик, как правило, отступало на второй план, так что характеристики даже основных алгоритмов анализа и синтеза автоматов почти не были изучены.

Эта статья посвящена исследованию сложности алгоритмов анализа и синтеза автоматов В.М. Глушкова. При этом основное внимание уделяется изучению переходов, определяемых диаграммой

$$G \to \mathbf{A}(G) \to \mathbf{S}(\mathbf{A}(G)),$$

где G — источник, $\mathbf{A}$, $\mathbf{S}$ — алгоритмы анализа и синтеза соответственно.

Будем рассматривать регулярные выражения над алфавитом $A = \{a_1, \ldots, a_m\}$, построенные при помощи операций $\vee$ (дизъюнкция), $\cdot$ (конъюнкция), $<>$ (итерация), а также символа e пустого слова. Регулярное событие, определяемое выражением R, обозначим $|R|$.

Регулярное выражение можно задавать также при помощи так называемых источников. Источник над алфавитом A представляет собой конечный ориентированный граф G, каждому ребру которого сопоставлен символ алфавита A, выделена вершина v_0, называемая начальной, выделено подмножество $F \neq \emptyset$ вершин, называемых финальными, причем выполнены условия:

а) различным ребрам, выходящим из одной вершины, сопоставлены различные символы алфавита A,

б) для каждой вершины v существует путь π_1, ведущий к v от v_0, и путь π_2, ведущий от v к некоторой финальной вершине.

Событие $|G|$, определяемое источником G, состоит из всех слов алфавита A, соответствующих путям, ведущим в источнике G от вершины v_0 к вершинам из F.

Задача анализа источника G состоит в построении регулярного выражения R, такого, что $|R| = |G|$. Задача синтеза состоит в построении по заданному регулярному выражению R такого источника G, что $|G| = |R|$. Предложенные В.М. Глушковым автоматные алгоритмы анализа и синтеза естественным образом переносятся на случай источников. Построенные при помощи этих алгоритмов анализа и синтеза регулярные выражения и источники обозначаем соответственно $\mathbf{A}(G)$ и $\mathbf{S}(R)$.

При решении задачи анализа с помощью алгоритма $\mathbf{A}$ возникает задача описания множества $\overline{R} = \{R : R = \mathbf{A}(G)\}$. Ее решение позволяет установить точное соответствие между классами источников J и соответствующими регулярными выражениями, восстанавливаемыми по источнику с помощью алгоритма $\mathbf{A}$. Полученное описание позволяет перейти к задаче синтеза, т.е., задаче описания всех источников J', получаемых из регулярных выражений множества $\overline{R}$ с помощью алгоритма Глушкова. На этом этапе основной задачей является описание класса всех источников, возникающих указанным образом. При этом представляется важным соотношение как классов J и J', так и их частей. Здесь приведем точное конструктивное описание класса $\overline{R}$ и класса J', а также покажем, в каких случаях $G \in J$ и соответствующий $G' \in J'$ изоморфны, т.е. практически совпадают.

Определим некоторые понятия, связанные с регулярными выражениями.

Дадим индуктивное определение *ветви регулярного выражении* над алфавитом:

1°. Ветвью пустого слова e является само это слово.

2°. Для каждой буквы a алфавита A ее ветвями являются пустое слово и сама буква a.

3°. Если $P_1 \vee \cdots \vee P_n$ — регулярное выражение, то его ветвями являются ветви выражений $P_1, \ldots, P_n$.

4°. Ветвями регулярного выражения $R = P_1 \ldots P_k$ являются всевозможные выражения вида $P_1 \ldots P_l P$, где P — ветвь выражения P_{l+1} $(0 \leq l \leq k-1)$.

5°. Ветвями регулярного выражения $R =< P >$ служат слово e и всевозможные выражения вида $< P > L$, где L — ветвь выражения P.

Пусть $R < P$ означает, что регулярное выражение R является ветвью регулярного выражения P.

Регулярное выражение P будем называть *отделимым* от регулярного выражения Q (и обозначать $P \perp Q$), если P представим в виде LaM, где $La \not< Q$ и существует b, такое, что $Lb < Q$. P *слабо отделимо* от Q $(P \pm Q)$, если $P \neq Q$ и либо $P < Q$, либо $Q < P$, либо $P \perp Q$ (L, M — регулярные выражения, a, b — буквы).

Пусть регулярное выражение R имеет вид

$$P_0 < Q_1 > P_1 \ldots P_{m-1} < Q_m > P_m \quad (m \geq 0),$$

где P_i $(1 \leq i \leq m)$ — слова алфавита A, $P_i \neq e$ $(l \leq i \leq m-1)$. Если для любого $i \in \{1, 2, \ldots, m\}$ выражение $R_i = P_i < Q_{i+i} > P_{i+1} \cdots < Q_m > P_m$ слабо отделимо от выражения Q_i, то назовем R *членом* 1*-го типа*. Если, кроме того, $P_0 \neq e$, $P_m \neq e$ и для любого $i \in \{1, 2, \ldots, m\}$ выражение R_i отделимо от выражения Q_i, то назовем R *членом* 2*-го типа*.

Следующая теорема позволяет описать множество $\overline{R}$ всех регулярных выражений, получающихся способом $\mathbf{A}$ анализа источников.

Теорема 1. *Регулярное выражение R принадлежит множеству $\overline{R}$ тогда и только тогда, когда выполняются условия:*

1) *каждый член выражения R есть член* 1*-го типа, слабо отделимый от любого другого члена R;*

2) *если в R входит выражение вида $< P >$, то каждый член P есть член* 2*-го типа, отделимый от любого другого члена P.*

Пусть G — источник над алфавитом A; $R = \mathbf{A}(G)$ — регулярное выражение, построенное способом $\mathbf{A}$ анализа источника G; $G' = \mathbf{S}(\mathbf{A}(G))$ — источник, построенный способом $\mathbf{S}$ синтеза источников по выражению $R = \mathbf{A}(G)$. В связи с Теоремой 1, дающей описание множества $\overline{R}$, естественным образом возникают вопросы об описании множества источников, получающихся в результате применения алгоритма синтеза $\mathbf{S}$ В.М. Глушкова к выражениям из $\overline{R}$, а также взаимосвязи источников G и $\mathbf{S}(\mathbf{A}(G))$ Для формулировки теорем, которые дают ответ на эти вопросы, нам понадобится ввести следующие два понятия.

Путь $\pi = v_0\rho_1 v_1 \dots \rho_l v_l$ в источнике G с начальной вершиной v_0, где $v_j \neq v_r$, $j \neq r$; $r \in \{1, 2, \dots, l-1\}$, $v_j \neq v_0$, $j \in \{1, 2, \dots, l-1\}$, $l \geq 0$, будем называть *полупростым путем*.

Пусть $\pi = v_0\rho_1 v_1 \dots \rho_s v_s$ — полупростой путь в источнике G; ρ — ребро выходящее из v_s и ведущее к некоторой вершине v. *Приведенный путь* $\overline{\pi\rho}$ пути $\pi\rho$ определим следующим образом:

а) если $v_s \neq v_i$ при всех i $(0 \leq i \leq s-1)$, то $\overline{\pi\rho} = \pi\rho$;

б) если $v_s \neq v_i$ $(0 \leq i \leq s-1)$, то $\overline{\pi\rho} = v_0\rho_1 v_1 \dots \rho_i v_i \rho v$.

Очевидно, что $\overline{\pi\rho}$ — снова полупростой путь.

Теорема 2. *Пусть G — источник, $G' = \mathbf{S}(\mathbf{A}(G))$. Тогда G' изоморфен источнику $\overline{G}$, определяемому по G следующим образом:*

а) *вершинами источника $\overline{G}$ служат полупростые пути источника G, причем начальной вершиной — пустой путь, а финальными вершинами — пути, ведущие к финальным вершинам источника G;*

б) *если π — полупростой путь в G, ведущий к вершине v, причем из вершины v выходит ребро ρ с отметкой a, то из вершины π источника $\overline{G}$ проводится ребро ρ к вершине $\overline{\pi\rho}$.*

Способы $\mathbf{A}$ анализа источников и $\mathbf{S}$ синтеза источников назовем *согласованными* на некотором источнике G, если источник $\mathbf{S}(\mathbf{A}(G))$ изоморфен источнику G.

Следующая теорема отвечает на вопрос о том, когда способы $\mathbf{A}$ анализа и $\mathbf{S}$ синтеза источников являются согласованными.

Теорема 3. *Способы $\mathbf{A}$ анализа источников и $\mathbf{S}$ синтеза источников являются согласованными на источнике G тогда и только тогда, когда источник G — дерево с корнем в начальной вершине.*

Теперь оценим усложнение $r(G) = ||G'||/||G||$ источника $G' = \mathbf{S}(\mathbf{A}(G))$ по сравнению с источником G; $||G||$ означает число вершин источника G. Имеет место следующая теорема

Теорема 4. *Пусть $r(n) = \max_{||G||=n} r(G)$; $\overline{r}(n) = \min_{||G||=n} r(G)$ Тогда*

$$r(n) = \begin{cases} \dfrac{m^{n+1}-1}{n(m-1)}, & m > 1, \\ (n+1)/n, & m = 1, \end{cases} \quad , \quad \overline{r}(n) = 1.$$

Займемся теперь оценкой сложности регулярных выражений, принадлежащих множеству $\overline{R}$ всех регулярных выражений, получающихся способом $\mathbf{A}$ анализа источников.

Определим *сложность* $||R||$ регулярного выражения R в алфавите $A = \{a_1, \dots, a_m\}$:

$$||e|| = 0;$$
$$||a_i|| = 1, \quad a_i \in A;$$
$$|| <R> || = ||R|| + 1;$$
$$||(R_1 \vee R_2)|| = ||(R_1 \cdot R_2)|| = ||R_1|| + ||R_2|| + 1.$$

Введем функцию Шеннона

$$L_{\mathbf{A}}(n) = \max_{||G||=n} ||\mathbf{A}(G)||,$$

где G — источник, $\mathbf{A}(G)$ — регулярное выражение, принадлежащее $\overline{R}$, и $||\mathbf{A}(G)||$ — сложность выражения $\mathbf{A}(G)$.

Нижняя и верхняя оценки величины $L_{\mathbf{A}}(n)$ даются в следующей теореме.

Теорема 5. *Имеют место неравенства*

$$2m^{n^2/8} \leq L_{\mathbf{A}}(n) \leq 4m^{n^2/2+5n/2} \quad (m > 1).$$

Важно отметить, что может быть указан алгоритм $\mathbf{A}_1$ анализа источников, который представляет собой некоторую модификацию алгоритма $\mathbf{A}$ [2]. Для алгоритма $\mathbf{A}_1$ анализа источников рассмотрим задачу, аналогичную задаче, рассматриваемой в предыдущей теореме. Пусть G — источник,

$R = \mathbf{A}_1(G)$ — регулярное выражение, построенное способом анализа источников и

$$L_{\mathbf{A}_1}(n) = \max_{||G||=n} ||\mathbf{A}_1(G)||,$$

где max берется по всем источникам G сложности n. В следующей теореме даются верхняя и нижняя оценки функции $L_{\mathbf{A}_1}(n)$.

Теорема 6. *Имеют место следующие неравенства*

$$2m^n \le L_{\mathbf{A}_1}(n) \le 5m^{3n} \quad (m > 1).$$

Последнее утверждение показывает, что величина $L_{\mathbf{A}_1}(n)$ существенно меньше, чем $L_{\mathbf{A}}(n)$ т. е., что небольшое изменение алгоритма $\mathbf{A}$ приводит к алгоритму, дающему выражения гораздо меньшей сложности. В этом смысле можно сказать, что алгоритм $\mathbf{A}$, предложенный В. М. Глушковым, неоптимальный.

Как известно [2], произвольное регулярное событие может быть представлено различными регулярными выражениями. Отсюда возникает задача представления его выражением минимальной сложности. Пусть $R(G)$ — регулярное выражение минимальной сложности, такое, что $|R(G)| = |G|$; обозначим

$$L(n) = \max_{||G||=n} ||R(G)||.$$

Если $\mathfrak{R}$ — произвольный алгоритм анализа, то функция

$$L_{\mathfrak{A}}(n) = \max_{||G||=n} ||\mathfrak{A}(G)||.$$

рассматриваемая в качестве меры эффективности алгоритма $\mathfrak{A}$, должна мажорировать функцию $L(n)$. Поведение функции $L(n)$ может быть охарактеризовано при помощи следующего утверждения.

Теорема 7. *При $m \ge 2$ имеет место оценка*

$$2[m/2]([n/2])^2 - 1 \le L(n) \le 5m^{3n}.$$

Из последней теоремы, в частности, следует, что никакое улучшение способа анализа $\mathbf{A}$ не позволяет строить выражения, сложность которых зависела бы линейно от сложности соответствующих источников.

При использовании сложности алгоритмов анализа автоматов естественно выделять такие классы источников, применение к которым исследуемого алгоритма дает либо только минимальные, либо, наоборот, только неминимальные регулярные выражения. Аналогичный подход естественно применять при исследовании сложности алгоритмов синтеза. Можно сказать, что здесь речь идет в определенном смысле об оптимальности рассматриваемых алгоритмов анализа и синтеза автоматов.

Обозначим через K_n множество источников над алфавитом A, имеющих не более чем n вершин; K'_n — множество всех таких источников G из K_n, что регулярное выражение $\mathbf{A}(G)$ минимально. Посредством $|K|$ будем обозначать число источников в классе K.

Следующее утверждение показывает, что "почти всегда" регулярное выражение, построенное при помощи алгоритма $\mathbf{A}$, не минимально.

Теорема 8. *При $n \to \infty$*

$$\frac{|K'_n|}{|K_n|} \to 0.$$

Вместе с приведенным утверждением можно эффективно указать два бесконечных подкласса K_1 и K_2 источников, применение к которым алгоритма $\mathbf{A}$ дает, соответственно, только минимальные и только не минимальные выражения.

Пусть K_1 — класс источников вида, указанного на Рис. 1, где $a_{ir_i} \ne a_{jr_j}$ при $i \ne j$, v_0 — начальная и v_f — финальная вершина. Пусть также K_2 — класс источников, имеющих более чем m финальных вершин (m — число букв алфавита A).

Теорема 9. а) *Для любого источника $G \in K_1$ выражение $\mathbf{A}(G)$ — минимально.*

б) *Для любого источника $G \in K_2$ выражение $\mathbf{A}(G)$ — не минимально.*

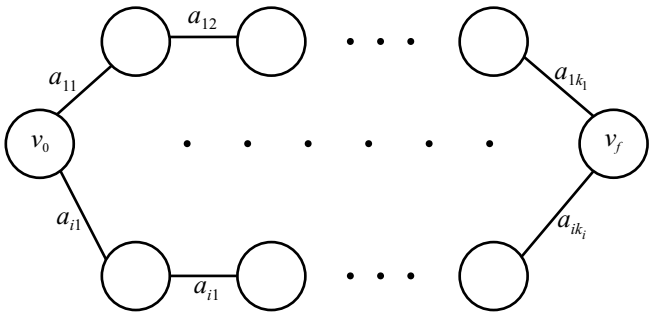


Рис. 1

Ситуация, аналогичная Теореме 9, возникает и для улучшенного способа синтеза $\mathbf{S}(\pi, \mathcal{D})$ В.М. Глушкова, описанного в [3]. Перед тем как перейти к исследованию эффективности алгоритма синтеза $\mathbf{S}(\pi, \mathcal{D})$, введем ряд новых понятий.

Пусть v — произвольная вершина источника G; G_v — источник, получающийся из источника G выбором вершины v в качестве начальной и удалением всех вершин источника G, которые недостижимы из вершины v, вместе с инцидентными им ребрами. Различные вершины v_1 и v_2 источника G будем называть *эквивалентными*, $v_1 \sim v_2$, если $|G_{v_1}| = |G_{v_2}|$.

Источник G над алфавитом A назовем *минимальным*, если не существует источника G' над A с меньшим, чем у G, числом вершин, и такого, что $|G| = |G|'$.

Нетрудно установить, что источник G минимален тогда и только тогда, когда любые две его различные вершины v_1 и v_2 не эквивалентны.

Слова p и q будем называть *циклически подобными*, если найдутся такие слова r_1 и r_2, что $p = r_1^k$, $q = r_2^l$, причем слова r_1 и r_2 получаются друг из друга циклическим сдвигом (k, l — натуральные).

Пусть $\mathcal{M}$ класс всех принадлежащих $\overline{R}$ регулярных выражений вида

$$\mathcal{D} = \mathcal{D}_1 \vee \cdots \vee \mathcal{D}_m,$$

где $\mathcal{D}_i = \mathcal{P}_{i0} < Q_{i1} > \mathcal{P}_{i1} \ldots \mathcal{P}_{i,s_i-1} < Q_{is_i} > \mathcal{P}_{is_i}$ $(1 \le i \le m)$, $s_i \ge 0$, $\mathcal{P}_{ij}$, Q_{ij} $(1 \le j \le s_i)$ — слова, $Q_{ij} \ne e$ и $\mathcal{P}_{ij} \ne e$ при $j \ne 0, s_i$, последняя буква слова $\mathcal{P}_{ij}$ отличается от последней буквы слова $Q_{i,j+1}$ $(0 \le j \le s_i - 1)$ и начальная буква слова $\mathcal{P}_{ij}$ отличается от начальной буквы слова Q_{ij} $(0 \le j \le s_i)$.

Обозначим через $\mathcal{M}_1$ подкласс класса $\mathcal{M}$ регулярных выражений, у которых слова Q_{is_i} не являются циклически подобными.

Пусть $\mathcal{M}_2$ — подкласс класса $\mathcal{M}$, такой, что существуют $k, l \in \{1, 2 \ldots, m\}$ такие, что $\mathcal{D}_l = R_l < Q_{ls_l} > \mathcal{P}_{ls_l}$, $\mathcal{D}_k = R_k < Q_{ls_l} > \mathcal{P}_{ls_l}$, причем кратчайшее слово события $|R_t|$ $(t \in \{k, l\})$ не является началом ни одного слова события $\mathcal{D}_r$ при $r \ne t$ и $|Q_{ls_l}| \ge 2$ ($|Q_{ls_l}|$ — длина слова Q_{ls_l}).

Теорема 10. а) *Для каждого регулярного выражения $\mathcal{D} \in \mathcal{M}_1$ существует такая последовательность отождествлений π, что источник $\mathbf{S}(\pi, \mathcal{D})$ — минимален.*

б) *Для любого выражения $\mathcal{D} \in \mathcal{M}_2$ и последовательности отождествлений π источник $\mathbf{S}(\pi, \mathcal{D})$ — не минимален.*

Доказательство указанных теорем опирается на ряд вспомогательных утверждений, некоторые из которых представляют самостоятельный интерес.

Список литературы

1. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. — М.: Наука, 1985.

2. Глушков В. Б. Синтез цифровых автоматов. — М.: Физматиз, 1962.

3. Кудрявцев В. Б., Подколзин А. С., Ушчумлич Ш. Введение в теорию абстрактных автоматов. — М.: Изд-во МГУ, 1985.

Обобщенный алгоритм адаптивной морфологической фильтрации изображений [1]

И. И. Фаломкин,

Физический факультет МГУ им. М.В.Ломоносова, кафедра компьютерных методов физики,
e-mail:igor.falomkin@gmail.com

Алгоритм адаптивной морфологической фильтрации [3] был разработан для решения задачи локализации неизвестного объекта на изображении, при известном, с точностью до условий регистрации, изображении фона. Предложенный в данной работе алгоритм является его обобщением и позволяет решать различные задачи обработки и анализа изображений. Алгоритм адаптивной морфологической фильтрации основан на работах Ю.П.Пытьева [1,2] по морфологическим методам анализа изображений, и является объединением идей морфологической фильтрации изображений [4] и наращивания области [5]. Эффективность предложенного алгоритма продемонстрирована на примере решения следующих задач:

(A) Задача подавления аддитивного шума на изображении из класса кусочно–полиномиальных по яркости изображений;

(B) Даны изображение некоторой исходной сцены (изображение фона), и изображение сцены, полученной из исходной добавлением/удалением объектов и/или изменением условий регистрации (анализируемое изображение). Требуется найти ту часть поля зрения, на которой анализируемое изображение «искажено» изменениями в сцене;

(C) Задача сегментации текстурнозначного изображения.

Отметим, что частным случаем задачи B является задача локализации неизвестного объекта на изображении, при заданном, с точностью до условий регистрации, изображении фона.

Обычно для решения приведенных задач применяют принципиально различающиеся методы и алгоритмы. Например, для решения задачи подавления шума может применяться алгоритм линейной фильтрации, описанный в работе [7], а для решения задачи локализации — алгоритм ранговых корреляций [6]. Предложенный алгоритм позволяет решать приведенные задачи в рамках единого подхода. Проведенный вычислительный эксперимент показал, что предложенный алгоритм позволяет получить решения более качественные, чем решения, получаемые при помощи известных автору специализированных алгоритмов.

Введем необходимые обозначения. Полем зрения X назовем ограниченное подмножество множества $\mathcal{N}^2$, где $\mathcal{N}$ — множество натуральных чисел. Изображением $f(\cdot)$ назовем функцию, определенную на поле зрения X и принимающую конечные значения в $\mathcal{R}^1$. Значение $f(x)$ будем интерпретировать как яркость изображения $f(\cdot)$ в точке $x \in X$. Класс всех изображений, определенных на X, обозначим $\mathfrak{L}(X)$.

Центральное место в предложенном алгоритме занимает понятие свойства изображения.

Определение 1. *Свойством изображения назовем функцию $S(\cdot)$, заданную на множестве $\mathfrak{L}(X)$ всех изображений и принимающую значения* 0 *или* 1.

Другими словами, $S(\cdot)$ — индикаторная функция изображений, обладающих свойством $S(\cdot)$. Будем говорить, что изображение $f(\cdot) \in \mathfrak{L}(X)$ обладает свойством $S(\cdot)$, если выполняется равенство $S(f) = 1$. В противном случае будем говорить, что изображение $f(\cdot)$ свойством $S(\cdot)$ не обладает. Множество всех изображений из $\mathfrak{L}(X)$, обладающих свойством $S(\cdot)$, обозначим $\mathcal{V}$:

$$\mathcal{V} = \{f(\cdot) \in \mathfrak{L}(X) : \ S(f) = 1\}.$$

Задать свойство изображения можно задав множество изображений, им обладающих. Например, свойство изображения «постоянное по яркости» задается множеством

$$\mathcal{V}^{const} = \{f(\cdot) \in \mathfrak{L}(X) \ : \ f(x) = c, \ -\infty < c < +\infty, \ x \in X\}.$$

Пусть $\mathcal{A} = (A_1, \ldots, A_n)$ — некоторое разбиение поля зрения X, а $f^i(\cdot) \in \mathcal{V}$, $i = 1, \ldots, n$, — некоторый набор изображений, обладающих свойством $S(\cdot)$. Для анализа предъявляется изображение

$$g(x) = \sum_{i=1}^{n} f^i(x)\chi_{A_i}(x) + \nu(x), \quad x \in X, \tag{1}$$

[1]Работа выполнена при финансовой поддержке РФФИ, грант №05-01-00615.

где $\chi_{A_i}(\cdot) \in \mathfrak{L}(X)$ — индикаторная функция множества A_i, $i = 1, \ldots, n$, а случайное изображение $\nu(\cdot) \in \mathfrak{L}(X)$ моделирует помеху. Причем множества A_i и изображения $f^i(\cdot)$, $i = 1, \ldots, n$, априори не известны. Требуется построить оценку $\widehat{\mathcal{A}} = (\widehat{A}_1, \ldots, \widehat{A}_n) \in \mathcal{G}(X)$ неизвестного разбиения $\mathcal{A} = (A_1, \ldots, A_n)$ по предъявленному изображению $g(\cdot)$.

Можно показать, что частным случаем данной задачи являются задачи A, B и C, при выборе соответствующих свойств изображения. Например, в задаче A свойством является представимость изображения в виде кусочно–полиномиальной функции, заданной на поле зрения. В задаче B интересующим нас свойством является равенство анализируемого изображения и изображения фона по форме[1,2].

Рассмотрим обобщенный алгоритм адаптивной морфологической фильтрации, предложенный для решения задачи построения оценки $\widehat{\mathcal{A}}$. Пусть $\mathcal{D}$ — некоторый набор подмножеств поля зрения. Будем считать, что предъявляемые для анализа изображения удовлетворяют следующему требованию: каждое множество A_i таково, что для каждого x из A_i найдется хотя бы одно множество из $\mathcal{D}$, покрывающее x и содержащееся в A_i, $i = 1, \ldots, n$. Набор всех множеств из $\mathcal{D}$, покрывающих $x \in X$, обозначим $\mathcal{D}_x$. Для каждой точки $x \in X$ поля зрения выберем то множество из $\mathcal{D}_x$, на котором анализируемое изображение наиболее точно приближается изображением, обладающим свойством $S(\cdot)$. Выбранное множество обозначим $O_x \in \mathcal{D}_x$. Выбранная таким образом окрестность точки x имеет наибольший, среди всех множеств из $\mathcal{D}_x$, шанс целиком принадлежать тому из множеств A_i, $i = 1, \ldots, n$, к которому принадлежит точка x.

Пусть известна оценка $\widetilde{\mathcal{A}}^k$ множеств, составляющих неизвестное разбиение $\mathcal{A}$ на шаге k алгоритма. Найдем оценку $\widetilde{\mathcal{A}}^{k+1}$ на шаге $k+1$. Для каждого множества $\widetilde{A}_i^k$, найдем изображение $p_i^k(\cdot)$ из $\mathcal{V}$, наиболее точно приближающее анализируемое изображение на $\widetilde{A}_i^k$, $i = 1, \ldots, n$. Из всех точек, прилегающих к множеству $\widetilde{A}_i^k$, выберем точку $x_i^k \in X$, для которой анализируемое изображение $g(\cdot)$ наиболее точным образом приближается изображением $p_i^k(\cdot)$ на выбранной ранее окрестности, $i = 1, \ldots, n$. Использование в алгоритме некоторой окрестности анализируемых точек позволяет снизить влияние шума, а специальное правило выбора указанной окрестности позволяет исключить нежелательные эффекты (например, размытие) на границе между областями A_i, $i = 1, \ldots, n$. Среди всех точек x_i^k, $i = 1, \ldots, n$, выберем точку $x_{i(k)}^k \in X$, для которой приближение анализируемого изображения $g(\cdot)$ изображением $p_i^k(\cdot)$ на множестве $O_{x_i^k}$ наиболее точно, $i = 1, \ldots, n$. Оценки $\widetilde{A}_i^{k+1} \subset X$ множеств A_i на шаге $k+1$ построим по правилу: $\widetilde{A}_i^{k+1} = \widetilde{A}_i^k$, $i \in \overline{1,n}$, $i \neq i(k)$, $A_{i(k)}^{k+1} = A_{i(k)}^k \cup \{x_{i(k)}^k\}$. Оценки считаются построенными, если их объединение дает все поле зрения.

Начальные оценки $\widetilde{A}_i^0$, $i = 1, \ldots, n$, могут быть выбраны как исследователем, так и по некоторому алгоритму. Например, в качестве начальных оценок можно выбрать подмножества поля зрения, на которых анализируемое изображение наиболее точно приближается изображением, обладающим свойством $S(\cdot)$.

На рисунке 1 представлены типичные результаты решения предложенным алгоритмом задачи подавления шума на кусочно–полиномиальном изображении. На рисунке 2 можно видеть результат решения предложенным алгоритмом задачи сегментации текстурнозначного изображения. Под текстурнозначным изображением понимается случайное изображение с постоянными на каждом из множеств A_i, $i = 1, \ldots, n$, стохастическими характеристиками. На приведенном рисунке яркости анализируемого изображения в точках различных множеств A_i, $i = 1, \ldots, n$, распределены по одному закону, но имееют различные дисперсии.

Автор благодарит Пытьева Ю.П. за постановку интересной задачи и плодотворные обсуждения.

Список литературы

1. Пытьев Ю. П. Морфологический анализ изображений. //ДАН СССР, 1983, т. 269, №5, сс. 1061–1064.

2. Pyt'ev Yu. P. Morphological Image Analysis. //Pattern Recognition and Image Analysis, 1993, vol. 3, no. 1, pp. 19–28.

3. Фаломкин И. И., Пытьев Ю. П. Адаптивный морфологический алгоритм локализации объектов на изображении. // Математические методы распознавания образов. Доклады 12-й всероссийской конференции, 2005, сс. 463–466.

4. Животников И. И., Пытьев Ю. П., Фаломкин И. И. Об алгоритме фильтрации кусочно–постоянных изображений. // Интеллектуальные системы. Москва. в печати.

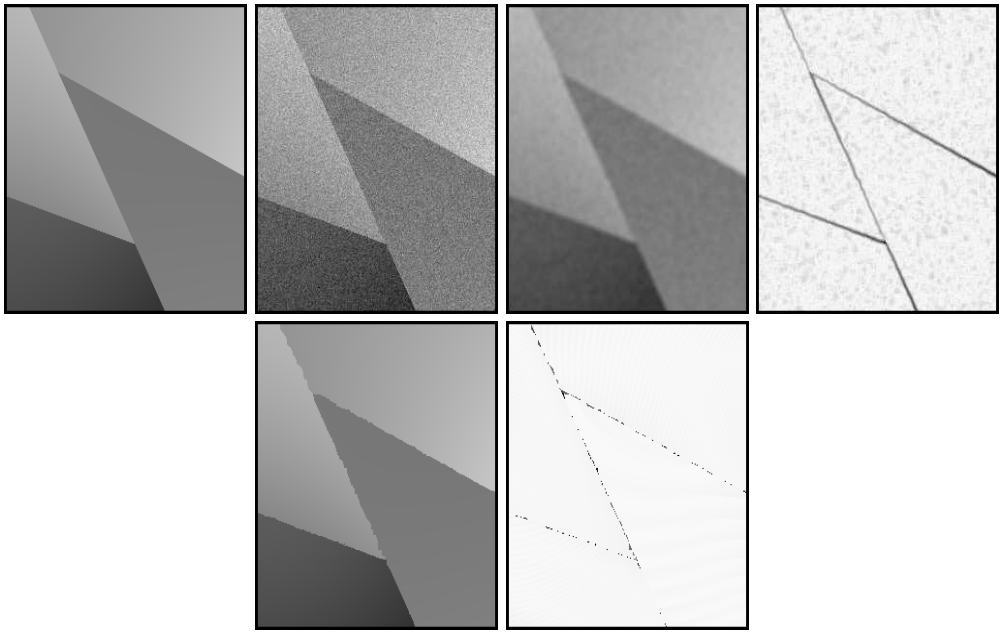


Рис. 1: Решение задачи шумоподавления. Представлены (слева направо): исходное изображение, зашумленное изображение, полученная оценка, невязка полученной оценки с исходным изображением. Верхняя «строка» — результат работы алгоритма линейной фильтрации. Нижняя — результат работы алгоритма адаптивной морфологической фильтрации.

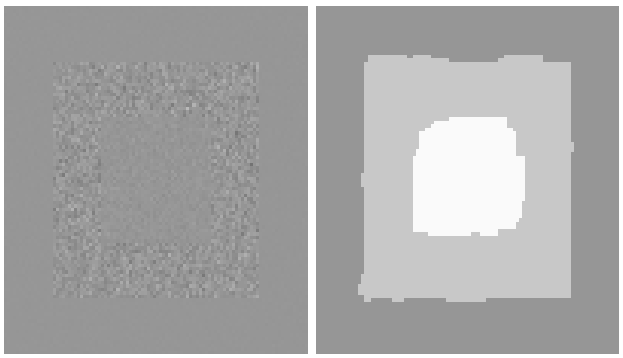


Рис. 2: Решение задачи сегментации текстурнозначного изображения. Представлены (слева направо): изображение, предъявленное для анализа, и результат работы алгоритма адаптивной морфологической фильтрации.

5. Adams R., Bischof L. Seeded region growing. //IEEE Trans. PAMI, Vol. 16, No. 6, pp. 641–647, 1994.

6. Кендэл М. Ранговые корреляции. -М.: Статистика, 1975.

7. Bow S. T. Pattern Recognition and Image Preprocessing. 2ed. Marcel Dekker Inc., 2002.

Псевдослучайные последовательности на основе INAR-модели и их свойства [2]

Харин Ю. С.,
чл.-корр. НАН Беларуси, зав. кафедрой математического моделирования и анализа данных
Белорусского государственного университета
E-mail: kharin@bsu.by
Ярмола А. Н.
факультет прикладной математики и информатики
Белорусского государственного университета
E-mail: and_ yarmola@tut.by

Введение

Генераторы псевдослучайных последовательностей являются неотъемлемой частью поточных криптосистем [**?**]. Такие криптосистемы используются в системах защиты информации, применяемых в компьютерных сетях, мобильной радиотелефонной связи и электронной торговле. Поэтому актуальной является задача построения псевдослучайных последовательностей (ПСП), удовлетворяющих требованию статистической близости распределений вероятностей фрагментов последовательности к равномерному распределению. Выделяются три основных подхода к построению алгоритмов генерации ПСП [1]: 1) прямые методы построения элементарных ПСП, к которым относятся конгруэнтные генераторы, генераторы Фибоначчи, линейные рекуррентные последовательности (ЛРП) в конечном поле; 2) методы «улучшения» элементарных ПСП, которые заключаются в специальных функциональных преобразованиях этих последовательностей; 3) методы комбинирования алгоритмов генерации. «Улучшение» элементарных ПСП преследует три цели: получение последовательности с более близкими к равномерному распределению распределений вероятностей фрагментов; увеличение периода последовательности; увеличение криптостойкости последовательности. Одним из основных подходов к «улучшению» ПСП является комбинирование ЛРП [2], которое осуществляется с использованием следующих операций: сложение, умножение, прореживание. В данном докладе в рамках этого подхода предлагается новый метод построения ПСП, основанный на использовании модели INAR(m) целочисленных временных рядов [3].

Определение INAR-генератора

Модель INAR(m) предложена Эльзаидом и Эль-Ошем [3] в качестве целочисленного аналога модели авторегрессии порядка m для решения задач анализа целочисленных экономических временных рядов $z_t \in \mathbf{Z}_+ = \{0, 1, 2, \ldots\}$, $t = 1, 2, \ldots$:

$$z_t = \sum_{j=1}^{m} p_j \circ z_{t-j} + \eta_t, \ t = m+1, m+2, \ldots, \tag{2}$$

где $\circ$ – оператор «биномиального прореживания» (binomial thinning): $p_j \circ z_{t-j} = \sum_{i=1}^{z_{t-j}} \xi_{t,i}^{(j)}$, $\{\xi_{t,i}^{(j)} : t = m+1, m+2, ..., \ i = 1, 2, ..., \ j = 1, ..., m\}$ – независимые в совокупности случайные величины Бернулли, $\mathbf{P}\{\xi_{t,i}^{(j)} = 1\} = 1 - \mathbf{P}\{\xi_{t,i}^{(j)} = 0\} = p_j$; $\{\eta_t\}$ – случайные величины, такие что η_t некоррелирована с $z_{t-1}, \ldots, z_{t-m}$, $t = m+1, m+2, \ldots$. С учетом этих обозначений модель (2) принимает вид:

$$z_t = \sum_{j=1}^{m} \sum_{i}^{z_{t-j}} \xi_{t,i}^{(j)} + \eta_t.$$

Используя эту идею модели INAR(m), построим генератор ПСП x_t следующем виде:

$$x_t = \Big(\sum_{i=1}^{m} \theta_i \sum_{j=0}^{x_{t-i}} \xi_{(t-1)N+j}^{(i)}\Big) \mod N, \ t = m+1, m+2, \ldots, \tag{3}$$

[2]Работа частично поддержана ГПФИ «Математические модели» (проект ММ-24).

где $\{\xi_t^{(i)} \in \{0,1\},\ i=0,\dots,m\}$ – выходные последовательности некоторых простейших (элементарных) бинарных генераторов $G_1,\dots,G_m$; в общем случае $\theta_j \in \mathcal{A}_N = \{0,1,\dots,N-1\}$, $\theta_m > 0$. Далее в этой статье исследуется специальный случай модели (3): $\theta_j \in \mathcal{A}_2 = \{0,1\},\ j=1,\dots,m-1$, $\theta_m = 1$; в этом случае величина θ_j определяет, подключен генератор G_j в данном сеансе генерации ПСП или нет. Отличительной особенностью генератора (3) среди известных генераторов ПСП является использование в (3) суммы случайного числа случайных величин.

Вероятностные свойства выходной последовательности INAR-генератора

При исследовании вероятностных свойств выходной последовательности INAR-генератора в качестве модели для выходных последовательностей $\{\xi_t^{(1)}\},\dots,\{\xi_t^{(m)}\}$ «элементарных» генераторов $G_1,\dots,G_m$ будем использовать последовательности независимых одинаково распределенных случайных величин Бернулли; также будем полагать что последовательности $\{\xi_t^{(1)}\},\dots,\{\xi_t^{(m)}\}$ – независимы. При этом уклонение $|p_j - 0,5| \in [0;0,5]$ характеризует «степень несовершенства» элементарного генератора G_j $(j=1,\dots,m)$. Обозначим: $r = \sum_{j=1}^m \theta_j$ – вес Хэмминга вектора коэфициентов $\theta = (\theta_1,\dots,\theta_m)'$; $1 \le k_1 < \ldots < k_r \le m \in$ – различные индексы, такие что $\theta_{k_j} = 1$, $j=1,\dots,r$, при этом $\theta_j = 0$ при всех других значениях j; $\mathrm{I}\{B\}$ – индикаторная функция события B.

Теорема 1. *Если $\{\xi_t^{(j)}\}$ – независимые в совокупности двоичные случайные величины, $\mathbf{P}\{\xi_t^{(j)} = 1\} = 1 - \mathbf{P}\{\xi_t^{(j)} = 0\} = p_j$, $j=1,\dots,m$, $t = m+1, m+2,\dots$, то последовательность x_t, определяемая (3), – однородная цепь Маркова порядка m.*

Из теоремы 1 следует, что в последовательности (3) появляются стохастические зависимости, которых не было в «элементарных» последовательностях, однако, данный факт не ухудшает вероятностные свойства последовательности (3), что будет показано в следующем разделе.

Пусть $P = (p_{i_0,\dots,i_m})$, $i_0,\dots,i_m \in \mathcal{A}_N$, – матрица вероятностей переходов цепи Маркова x_t, $p_{i_0,\dots,i_m} = \mathbf{P}\{x_t = i_m \mid x_{t-1} = i_{m-1},\dots,x_{t-m} = i_0\}$, $i_0,\dots,i_m \in \mathcal{A}_N$, $t > m$.

Лемма 1. *Если $\theta_1,\dots,\theta_m \in \mathcal{A}_2$, то элементы матрицы вероятностей переходов P имеют вид $(i_0,i_1,\dots,i_m \in \mathcal{A}_N)$*

$$p_{i_0,\dots,i_m} = \sum_{k=0}^{m-1} \mathbf{I}\{i_m + kN \le S(i_0,\dots,i_{m-1})\} P(i_m + kN, i_0,\dots,i_{m-1}). \tag{4}$$

$$\begin{aligned} P(i,i_0,\dots,i_{m-1}) = \sum_{j_1=0}^{\min(N_1,i)} \sum_{j_2=0}^{\min(N_2,i-j_1)} \cdots \sum_{j_{r-1}=0}^{\min(N_{r-1},i-j_1-\ldots-j_{r-2})} \prod_{l=1}^{r-1} \left(\mathrm{C}_{N_l}^{j_l} p_{k_l}^{k_l} (1-p_{k_l})^{N_l - j_l}\right) \times \\ \times \mathrm{C}_{N_r}^{i-j_1-\ldots-j_{r-1}} p_{k_r}^{i-j_1-\ldots-j_{r-1}} (1-p_{k_r})^{N_r - i + j_1 + \ldots + j_{r-1}}, \\ i = 0,\dots,S(i_0,\dots,i_{m-1}), \end{aligned} \tag{5}$$

$$S(i_0,\dots,i_{m-1}) = \sum_{j=1}^{m} \theta_j (i_{m-j} + 1),\ N_j = i_{m-k_j} + 1,\ j=1,\dots,r. \tag{6}$$

Следствие 1. *Если в условиях леммы 1 генераторы $\{G_j\}$ – однородные: $p_1 = p_2 = \ldots = p_m = p$, то $(i_0,\dots,i_m \in \mathcal{A}_N)$*

$$p_{i_0,\dots,i_m} = \sum_{k=0}^{m-1} \mathbf{I}\{i_m + kN \le S(i_0,\dots,i_{m-1})\} \mathrm{C}_{S(i_0,\dots,i_{m-1})}^{i_m+kN} p^{i_m+kN} (1-p)^{S(i_0,\dots,i_{m-1}) - i_0 - kN}. \tag{7}$$

Таким образом, вероятности переходов зависят только от i_m и величины $S(i_0,\dots,i_{m-1}) \in \{r, r+1,\dots,Nr\}$, и для вычисления матрицы вероятностей переходов необходимо определить только $(N-1)((N-1)r+1)$ значений вероятностей переходов.

Лемма 2. *Если $r \ge N-1$ и $0 < p_{k_j} < 1$, $j=1,\dots,r$, то цепь Маркова x_t – эргодическая.*

Вероятностные свойства для бинарного INAR-генератора

Исследуем наиболее важный на практике бинарный случай (3), когда $N=2$ и генераторы $G_1,\dots,G_m$ – однородные:

$$p_j = \frac{1}{2}(1+\varepsilon),\ j=1,\dots,m,\ |\varepsilon| < 1, \tag{8}$$

где $\varepsilon \in (-1;1)$ – величина, характеризующая «степень несовершенства» генераторов $G_1,\dots,G_m$ (если $\varepsilon = 0$, то $p_1 = \ldots = p_m = 1/2$ и все генераторы $G_1,\dots,G_m$ порождают «чисто случайные» последовательности). Отметим также, что в силу (8) и леммы 2, x_t – эргодическая цепь Маркова.

Обозначим $\Pi^{(t)} = \left(\pi^{(t)}_{i_1,\ldots,i_m}\right)$, $i_1,\ldots,i_m \in \mathcal{A}_2$, – m-мерное распределение вероятностей процесса x_t, $\pi^{(t)}_{i_1,\ldots,i_m} = \mathbf{P}\{x_{t-m+1} = i_1,\ldots,x_t = i_m\}$; $\Pi^* = \left(\pi^*_{i_1,\ldots,i_m}\right)$, $i_1,\ldots,i_m \in \mathcal{A}_2$, – m-мерное стационарное распределение вероятностей цепи Маркова [4], которое существует в силу установленного свойства эргодичности.

Исследуем распределение вероятностей цепи Маркова m-ого порядка x_t, порождаемой генератором (3), с помощью следующих функционалов, возникающий в задаче дискриминантного анализа цепей Маркова [5]: $\Delta = 2^{-m}\sum_{i_0,\ldots,i_m\in\mathcal{A}_2}|p_{i_0,\ldots,i_m} - 0,5| \geq 0$ – среднее уклонение распределения вероятностей одношаговых переходов от равномерного распределения; $Z_m = \sum_{i_1,\ldots,i_m\in\mathcal{A}_2}|\pi^*_{i_1,\ldots,i_m} - 2^{-m}| \geq 0$ – октаэдрическая норма уклонения стационарного m-мерного распределения Π^* от равномерного на $\mathcal{A}_2^m$ распределения. Чем меньше Δ, Z_m, тем выходная последовательность генератора (3) ближе к «чисто случайной последовательности».

Лемма 3. *Если $N = 2$ и выполнены условия (8), то для любых :*

$$p_{i_0,\ldots,i_{m-1},0} = p_{i_0,\ldots,i_{m-1},1} = 1/2\ \forall i_0,\ldots,i_{m-1} \in \mathcal{A}_2$$

тогда и только тогда, когда $p = 1/2$.

Из леммы 3 следует, что предложенный метод не позволяет получить «совершенный» генератор из «несовершенных», однако следующая теорема позволяет оценить количественно «степень несовершенства» генератора ПСП (3) и выяснить факторы, от которых она зависит.

Теорема 2. *Если $N = 2$ и выполнены условия (8), то*

$$p_{i_0,\ldots,i_m} = (1 + (-1)^{i_m}(-\varepsilon)^{S(i_0,\ldots,i_{m-1})})/2,\ i_0,\ldots,i_m \in \mathcal{A}_N, \tag{9}$$

$$\Delta = |\varepsilon/2|^r(1 + 2|\varepsilon|)^r. \tag{10}$$

Отметим, что для выходной последовательности «элементарного» генератора $\Delta = |\varepsilon|$; следовательно, в силу теоремы 2, несмотря на наличие в выходной последовательности (3) стохастических зависимостей, данная последовательность (3) является «менее предсказуемой», чем выходные последовательности «элементарных» генераторов $G_1,\ldots,G_m$.

Теорема 3. *Если выполнены условия (8), то справедливо неравенство: $Z_m \leq Z_+$, причем для верхней границы Z_+ справедливо асимптотическое ($\varepsilon \to 0$) разложение:*

$$Z_+ = 2^{-r}(1-\varepsilon)|\varepsilon|^r(1+|\varepsilon|)^{r-1} \times \left(1 + \sum_{j=1}^{m-1} 2^{2(r_j - j)}\frac{|\varepsilon|^{r_j}}{(1+|\varepsilon|)^{r_j}}\prod_{k=1}^{j}(1 + (-\varepsilon)^{\theta_{m-k}})\right) + \mathcal{O}(\varepsilon^{2r}), \tag{11}$$

где $r_j = \sum_{k=1}^{j}\theta_{m-k}$, $j = 1,\ldots,m-1$.

В силу теоремы 3, $Z_m = \mathcal{O}(|\varepsilon|^r)$; таким образом, распределение m-векторов (m-грамм) выходной последовательности INAR-генератора близко к равномерному на $\mathcal{A}_2^m$ распределению с погрешностью $\mathcal{O}(|\varepsilon|^r)$. Заметим, что для выходной последовательности «элементарного» генератора погрешность значительно больше: $Z_m = O(|\varepsilon|)$. Следовательно, распределение вероятностей фрагментов выходной последовательности (3) ближе к равномерному, чем распределение фрагментов такой же длины для «элементарного» генератора.

Список литературы

1. Харин Ю.С., Берник В.И., Матвеев Г.В., Агиевич С.В. Математические и компьютерные основы криптологии. Минск: Новое знание, 2003.

2. Варфоломеев А.А., Жуков А.Е., Пудовина М.А. Поточные криптосистемы. Основные свойства и методы анализа стойкости. Учебное пособие. М.: ПАИМС, 2000. - 272 с.

3. Alzaid, A.A. and Al-Osh, M. An integer-valued pth-order autoregressive structure (INAR(p)) process.// Journal of Applied Probability. 1990, 27, 314-324.

4. Боровков А.А. Теория вероятностей. - М.:Наука, 1986.

5. Kharin Yu., Kostevich A. Discriminant analysis of stationary finite Markov chains. // Math. Methods of Statistics. 2004, Vol. 13, No.1, pp. 235-252.

Логические элементы на нейронах[1]

Хачумов В. М.
зав. лабораторией интеллектуального управления ИЦИИ ИПС РАН
152020, г. Переславль - Залесский, ИПС РАН,
e-mail: vmh@vmh.botik.ru

Введение

Расматривается задача моделирования элементов вычислительной техники (ВТ) на искусственных нейронах и нейронных сетях (ИНС). Реализация функционально полного набора логических элементов на нейронах позволяет решать задачи моделирования сложных устройств без предварительной настройки ИНС. Например, она дает ключ к прозрачному решению, так называемой «проблемы XOR», позволяет моделировать арифметические устройства ВТ и элементы памяти. Эта же задача может быть успешна решена на универсальных ИНС с использованием специальных настроек. В работе исследуются возможности генетического алгоритма настройки ИНС на функции элементов ВТ с использованием минимального числа нейронов и связей.

1. Решение проблемы XOR на нейронах с пороговой функцией

Создадим базовый набор логических элементов на искусственных нейронах. Для этого удобно использовать в качестве функции активации единичный скачок или логистическую функцию (сигмоид), соответственно:

$$f(s)=\begin{cases}1, & \textit{если} \quad s\ge 0\\ 0, & \textit{если} \quad s<0\end{cases}, \quad f(s)=\frac{1}{1+e^{-\lambda s}}\ .$$

Рассмотрим нейроны с одним и двумя входами (рис.1).

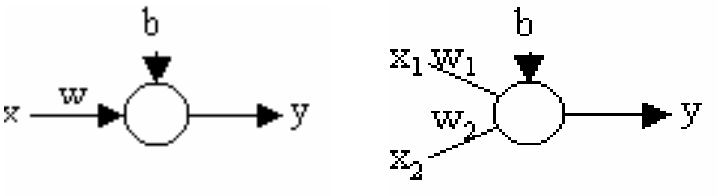


Рис.1. Нейроны с одним и двумя входами

Здесь b – смещение, w_1, w_2 - веса связей (весовые коэффициенты). Логические элементы создаем, назначая определенные весовые коэффициенты и смещения:

логический элемент «НЕ»:	$w=-1,\ b=0$ (элемент с одним входом).
логический элемент «И»:	$w_1=w_2=1,\quad b=-1.5;$
логический элемент «ИЛИ»:	$w_1=w_2=1,\quad b=-0.5;$
логический элемент «И-НЕ»	$w_1=w_2=-1,\ b=1.5;$
логический элемент «ИЛИ-НЕ»:	$w_1=w_2=-1,\ b=0.5;$

Количество возможных вариантов программных реализаций этих элементов бесконечно, однако в реальном техническом нейроне существуют ограничения на величины весов и подаваемых смещений. В данной работе эти ограниченния снимаются. Рассмотренный набор позволяет создавать на его основе более сложные логические схемы, моделировать вычислительные устройства и элементы памяти. Создадим ИНС, решающую проблему XOR. Проблема заключается в невозможности реализации функции XOR, соответствующей логическому выражению $y=x_1\bar{x}_2\vee\bar{x}_1x_2$, на одном нейроне. Этот

[1] Работа выполняется при поддержке РФФИ (проект N 06-07-89083)

известный факт используется в теории ИНС как демонстрация ограниченности возможности отдельного нейрона. Известны различные реализации XOR, называемой также «исключающее ИЛИ» и «сумма по mod 2», с использованием как пороговой так сигмоидальной функций активации. Можно реализовывать логическую функцию XOR на базе рассмотренных элементов, но тогда потребуется пять нейронов, что не рационально. Создадим дополнительно элемент, реализующий функцию $z = x_1\bar{x}_2$, Для этого полагаем: $w_1 = 1, w_2 = -1,\ b = -0.5$. На основе этого элемента и схемы «ИЛИ» получим реализацию XOR (рис.2). Здесь первый слой выполняет пребразование и переход к новой системе координат y_1, y_2. Промежуточные выходы определяются функциями $y_1 = x_1\bar{x}_2$, $y_2 = \bar{x}_1 x_2$. Нейрон второго слоя формирует разделяющюю линию $y_1 + y_2 - 0.5 = 0$, отраженную на рис.2b

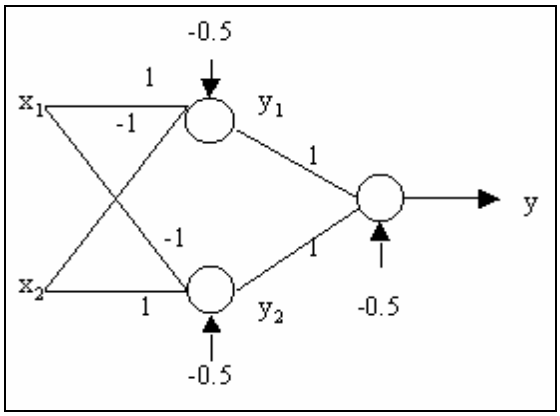


Рис.2a Схема XOR

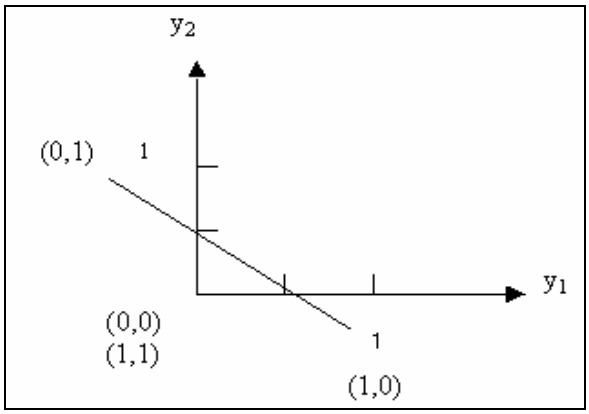


Рис.2 b. Разделяющая прямая

В отличие от примера [1], приведенного на рис.3, имеем симметричную структуру настроек и четкое логическое обоснование построения элемента. При этом уравнение разделяющей лини будет $y_1 - y_2 - 0.5 = 0$, как это показано на рис. 4б.

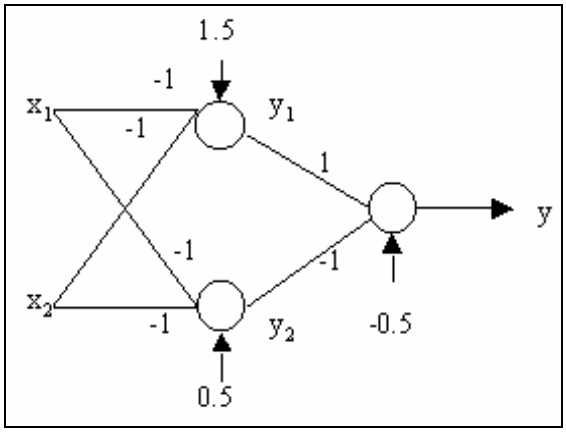


Рис.3a.Схема XOR [1]

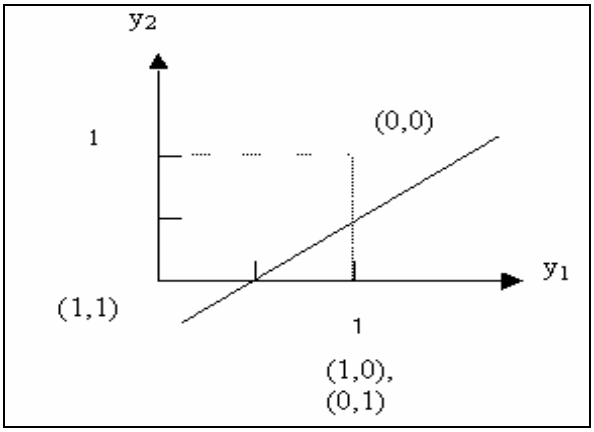


Рис.3b. Разделяющая прямая

2. Решение проблемы XOR на нейронах с сигмоидальной функцией

Использование сигмоидальной функции активации позволяет решить проблему XOR при одновременном уменьшении числа нейронов до двух [2]. В этом случае появляются дальние связи между входами и выходным нейроном. В настоящей работе настройка ИНС на функцию XOR выполнялась генетическим алгоритмом. Диапазон начальных значений: [-15,15]; число хромосом в популяции -100. На каждом шаге настройки для каждого из четырех входов (00, 01, 10, 11) определяется величина $f = \frac{1}{\delta + 1}$, где $\delta = \left| y_T - y_\Phi \right|$ – модуль разности между требуемым и фактическим выходами. Значением фитнес - функции является среднее арифметическое этих четырех величин. Чем полезнее хромосома, тем

ближе значение фитнес - функции к 1. Выбор двух хромосом для скрещивания проводился методом рулетки. Для скрещивания использовался метод обмена четных и нечетных элементов родительских хромосом, при этом в популяцию возвращалась пара наилучших хромосом. Настройка выполнялась за 1200-1300 итераций. Лучшая хромосома популяции: (6.68, -9.88, 12.1, -6.24, 14.6, -7.47, -2.98) со значением фитнес - функции: 0.999999. Среднее значение фитнес-функций улучшается с 0.742198 (на первом шаге) до 0.837396. Значение фитнес-функции лучшей хромосомы популяции растет с 0.875 до 0.999999. Результат настройки отражен на рис.4. Для сравнения на рис.5 показана реализация модели XOR, полученная в работе [2] по методу обратного распространения ошибки.

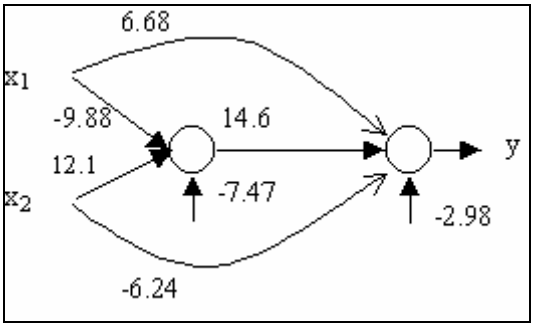


Рис. 4. Схема XOR (сигмоид)

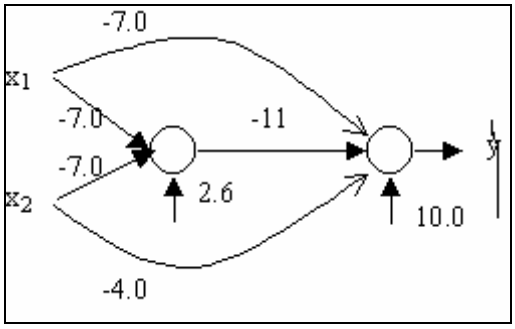


Рис. 5. Схема XOR [2]

3. Моделирование арифметических устройств на нейронах

Рассмотрим одноразрядный полусумматор, обозначаемый как SM/2. Он не имеет входного переноса, т.к. является самым младшим разрядом параллельного сумматора. Структура полусумматора, построенного на типовых элементах XOR и «И» приведена на рис.6.. В то же время представляет интерес задача построения суммирующего устройства как результата настройки универсальной ИНС прямого распространения. Структура ИНС для реализации полусумматора приведена на рис.7. Входной слой служит только для распределения сигналов, функция активации нейронов – сигмоид.

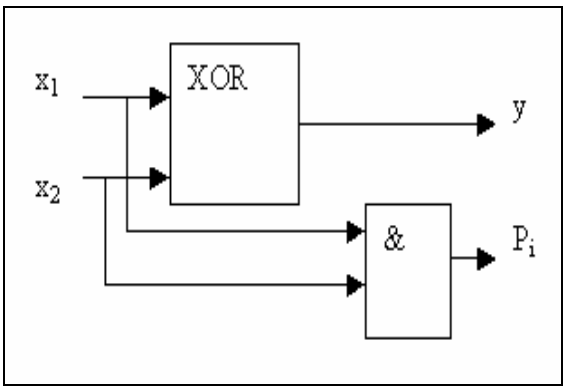


Рис.6. Логика SM/2

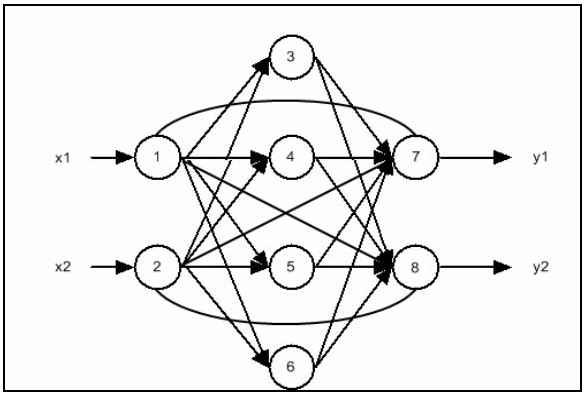


Рис.7. Структура ИНС для SM/2

На основе двух полусумматоров и схемы «ИЛИ» можно построить полный одноразрядный сумматор SM (рис.8). Для настройки ИНС был использован генетический алгоритм, изложенный ранее. Начальная популяция состояла из 16 хромосом, сгенерированных случайным образом. Гены (элементы хромосомы) назначались из диапазона [-17,17].

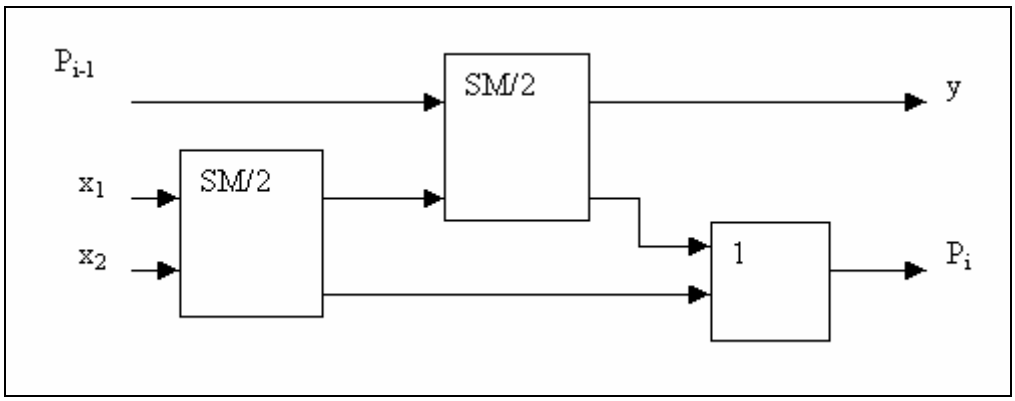


Рис.8. Полный одноразрядный сумматор SM

Алгоритм выполнял настройку сети за 900-4000 итераций, в зависимости от заданной точности, начального диапазона и выбора значений первоначальных настроек, задаваемых датчиком случайных чисел. Среднее значение фитнес-функций популяции улучшалось с 0.567667 (на первом шаге) до 0.997773. Значение фитнес-функции лучшей хромосомы популяции росло с 0.760721 до 0.99802. Полученные веса отражены в таблице.

	1	2	3	4	5	6	7	8
1	-	-	-9.72	-15.58	-6.81	-14.54	5.92	-6.77
2	-	-	4.07	2.61	-13.73	-10.93	-12.58	16.23
3	-9.72	4.07	-	-	-	-	13.47	-11.92
4	-15.58	2.61	-	-	-	-	6.4	-16.46
5	-6.81	-13.73	-	-	-	-	-16.75	-0.71
6	-14.54	-10.93	-	-	-	-	-16.42	6.41
7	5.92	-12.58	13.47	6.4	-16.75	-16.42	-	-
8	-6.77	16.23	-11.92	-16.46	-0.71	6.41	-	-

Таким образом, для построения модели SM/2 потребовалось 6 нейронов и 20 связей. Попытки настроить сеть с меньшим количеством нейронов и связей не привели к положительному результату. В то же время, реализация этого же устройства по схеме рис.6 требует всего 4 нейронов и 8 связей. Задача построения сумматора SM и других элементов ВТ на основе двухслойной ИНС (рис.7) представляет определенный интерес и должна решаться в сравнении с альтернативным подходом.

Заключение

В настоящей работе решена часть задачи, заключающейся в построении моделей типовых элементов ВТ на основе ИНС. Рассмотренные элементы (схема XOR, полусумматор) были реализованы на специализированных и универсальных двухслойных ИНС. Примененная схема генетического алгоритма позволила настраивать ИНС на заданные функции, но при относительно больших накладных расходах. Дальнейшая перспектива связана с расширением списка моделей элементов ВТ и использованием суперкомпьютера для ускорения процессов моделирования. Автор выражает благодарность студенту НОУ ИПС-УГП им. А. К. Айламазяна (г. Переславль-Залесский) Недеву М. Д., выполнившему программную реализацию алгоритмов.

Список литературы

1. Роберт Каллан. Основные концепции нейронных сетей. - М.: Издательский дом «Вильямс», 2001.

2. Джордж Ф. Люгер. Искусственный интеллект. Стратегии и методы решения сложных проблем. - М.: Издательский дом «Вильямс», 2003.- 864 с.

Возможен ли эффективный формальнологический вывод в нечетких моделях типа Мамдани?

Ходашинский И. А.
д.т.н., профессор кафедры автоматизации обработки информации
Томского государственного университета систем управления и радиоэлектроники
634050, Томск, пр. Ленина, 40, ТУСУР
hodashn@rambler.ru

Постановка задачи.

Пусть имеется объект исследования, заданный своей таблицей наблюдений. Проблема исследования обусловлена невозможностью построения аналитической модели изучаемого объекта, либо слишком большой сложностью такой модели, либо отсутствием достаточного опыта для построения экспертных систем, либо недостаточностью экспериментальных данных для статистического моделирования. Решением проблемы может быть переход от аналитических или статистических моделей к нечетким. Нечеткая модель выступает при этом в качестве универсального аппроксиматора [1] и может быть построена либо на основе неполных знаний эксперта, либо на основе ограниченных наблюдаемых данных.

Нечеткая модель определена как система с n входными переменными $\mathbf{X} = \{X_1,..., X_n\}$, заданными на входной области рассуждений $DX = DX_1 \times ... \times DX_n$, и одной выходной переменной Y, определенной на выходной области рассуждений DY. Четкое значения, которое принимает входная переменная X_i, обозначается как x_i, и как y для выходной переменной Y.

Нечеткая область определения i-ой входной переменной X_i, обозначена как $FX_i = \{LX_{i,1},..., LX_{i,pi}\}$, где pi – количество лингвистических термов (нечетких значений), на которых определена входная переменная, $LX_{i,k}$ задает функцию принадлежности и имя k-го лингвистического терма. Аналогично, $FY = \{LY_1,..., LY_q\}$ – нечеткая область определения выходной переменной, q – число нечетких значений, LY_l – функция принадлежности и имя выходного лингвистического терма.

Нечеткое j-ое правило обозначено как

$$R_j : LX_{1,j_1},...,LX_{n,j_n} \rightarrow LY_j \ . \qquad (1)$$

Для описания отображения входного вектора $\mathbf{X}$ в значение y используются методы нечеткой логики, например, аппроксимация Мамдани или метод, основанный на формальнологическом доказательстве [2]. В нечетком выводе участвуют операции конъюнкции и дизъюнкции. Задание этих операций на основе триангулярных норм (t-норм) позволяет более гибко настраивать нечеткую систему на исследуемую предметную область.

В общем случае процесс создания нечеткой модели состоит из следующих шагов:

1) определение входных и выходных переменных ($\mathbf{X}$, DX, Y, DY, pi, q);
2) определение логических операций на основе t-нормальных функций;
3) задание функций принадлежности каждой переменной (FX_i, FY);
4) определение нечетких правил (R_j);
5) определение типа нечеткого вывода.

Ниже рассмотрены некоторые основные этапы создания нечетких моделей.

Определение логических операций. Операции конъюнкции и дизъюнкции определяются через t-нормы и t-конормы (операторы T и S, соответственно), одним из важных свойств которых является двойственность по отношению друг к другу:

$$T_n(a_1, a_2, \ldots, a_n) = 1 - S_n((1-a_1, 1-a_2, \ldots, 1-a_n)),$$
$$S_n(a_1, a_2, \ldots, a_n) = 1 - T_n((1-a_1, 1-a_2, \ldots, 1-a_n)), \qquad (2)$$

при условии, что операция отрицания $\sim a$ задается как $1 - a$.

Ниже приведены некоторые t-нормальные и t-конормальные функции для n переменных.

Функции Заде:

$T(a_1, a_2, \ldots, a_n) = \min(a_1, a_2, \ldots, a_n)$, $S(a_1, a_2, \ldots, a_n) = \max(a_1, a_2, \ldots, a_n)$.

Вероятностные функции:

$T(a_1, a_2, \ldots, a_n) = a_1 * a_2 * \ldots * a_n$,

$$S(a_1, \ldots, a_n) = \left(\sum_{i=1}^{n} a_i - \sum_{i=1}^{n} \sum_{j>i}^{n} a_i a_j + \sum_{i=1}^{n} \sum_{j>i}^{n} \sum_{k>j}^{n} a_i a_j a_k \pm \ldots \pm \prod_{i=1}^{n} a_i \right).$$

Функции Лукасевича:

$$T(a_1, \ldots, a_n) = \max\left(\sum_{i=1}^{n} a_i - (n-1), 0 \right), \quad S(a_1, \ldots, a_n) = \min\left(\sum_{i=1}^{n} a_i, 1 \right).$$

Функции Швайцера и Скляра:

$$T(a_1, \ldots, a_n) = 1 - \left(\sum_{i=1}^{n} (1-a_i)^p - \sum_{i=1}^{n} \sum_{j>i}^{n} (1-a_i)^p (1-a_j)^p + \sum_{i=1}^{n} \sum_{j>i}^{n} \sum_{k>j}^{n} (1-a_i)^p (1-a_j)^p (1-a_k)^p \pm \ldots \pm \prod_{i=1}^{n} (1-a_i)^p \right)^{1/p}$$

$$S(a_1, \ldots, a_n) = \left(\sum_{i=1}^{n} a_i^p - \sum_{i=1}^{n} \sum_{j>i}^{n} a_i^p a_j^p + \sum_{i=1}^{n} \sum_{j>i}^{n} \sum_{k>j}^{n} a_i^p a_j^p a_k^p \pm \ldots \pm \prod_{i=1}^{n} a_i^p \right)^{1/p}.$$

В нечетких системах операция импликации классически определяется через операцию дизъюнкции или через функцию *t*-конормы:

$I(a_1, a_2) = S(1 - a_1, a_2)$.

Ниже перечислены три наиболее часто применяемые задания импликации *S*-типа.

1. *Импликация на функциях Заде*: $I(a_1, a_2) = S(1-a_1, a_2) = \max(1- a_1, a_2)$.
2. *Импликация на вероятностных функций*: $I(a_1, a_2) = 1- a_1 + a_2 * a_1$.
3. *Импликация Лукасевича*: $I(a_1, a_2) = \min(1- a_1 + a_2, 1)$.

Возможно задание импликации *T*-типа, которое основано на *t*-нормальной функции. Ниже приведено два способа задания таких импликаций:

Импликация по Мамдани: $I(a_1, a_2) = \min(a_1, a_2)$.

Импликация по Ларсену: $I(a_1, a_2) = a_2 * a_1$.

Тип нечеткого вывода.

В операторной форме нечеткое правило (1) при поступлении на вход $(x_1,\ldots, x_n)$ представляется следующим образом:

$R_j(x_1,\ldots, x_n; y) = I(T(LX_{1,j1}(x_1), \ldots, LX_{n,jn}(x_n)), LY_j(y))$.

Если импликация задана классическим способом (*S*-типа), то правило будет переписано следующим образом:

$R^s_j(x_1,\ldots, x_n; y) = S`(S(1-LX_{1,j1}(x_1), \ldots, 1-LX_{n,jn}(x_n)), LY_j(y))$.

Заметим, что если *t*-конормальная функция является параметрической, например, функция Швайцера-Скляра, то следует различать *t*-конормальные функции, определяющие импликацию и конъюнкцию (дизъюнкцию), отсюда штрих в обозначении оператора *t*-конормальной функции.

Если импликация *T*-типа, то правило будет иметь следующий вид:

$R^t_j(x_1,\ldots, x_n; y) = T`(T(LX_{1,j1}(x_1), \ldots, LX_{n,jn}(x_n)), LY_j(y))$.

Выбор оператора агрегации правил связан с выбором оператора импликации. Если оператор импликации определен классическим формальнологическим методом через *t*-конормальную функцию, то оператор агрегации правил определяется через *t*-нормальную функцию:

$R_{KL}(x_1,\ldots, x_n; y) = T`(R_1^s(x_1,\ldots, x_n; y), R_2^s(x_1,\ldots, x_n; y), \ldots, R_m^s(x_1 \ldots, x_n; y))$.

Если оператор импликации определен через *t*-нормальную функцию, то оператор агрегации определяется через *t*-конормальную функцию:

$R_M (x_1,..., x_n; y) = S`(R_1^t(x_1,..., x_n; y), R_2^t(x_1,..., x_n; y),..., R_m^t(x_1,..., x_n; y))$.

Указанный способ объединения правил совпадает с подходом, основанным на аппроксимации Мамдани [2].

Если на вход нечеткой модели поступают четкие значения (x_1, x_2,..., x_n), то с учетом того, что x_k является синглетоном соответствующего нечеткого множества, нечеткое выходное значение находится по формуле:

$F(y) = R(x_1,..., x_n; y)$.

В системе аппроксимации Мамдани нечеткое выходное значение вычисляется следующим образом:

$F_M(y) = R_M (x_1,..., x_n; y)$,

а нечеткий выход, основанный на формальнологическом подходе:

$F_{KL} (y) = R_{KL} (x_1,..., x_n; y)$.

Эксперимент

Невозможность проведения аналитических исследований нечетких систем моделирования заставляет обратиться к экспериментальным исследованиям. Нечеткий вывод рассмотрим на примере нечеткой системы оценивания величин [3,4]. Суть нечеткологического оценивания заключается в сопоставлении данных из двух разнотипных шкал: абсолютной нечеткой статической и сравнительной нечеткой статической. Задача оценивания имеет следующую формулировку: по известным абсолютной статической оценке величины A и сравнительной статической оценке величин A и B найти абсолютную статическую оценку величины B. Результатом такого сопоставления будет значение величины, выраженное в абсолютной нечеткой статической шкале. Система оценивания величин может рассматриваться как нечеткая система типа два_входа-один_выход, в которой правило имеет следующий вид:

Правило i: *ЕСЛИ* $АНС_1 = A_{1i}$ *И* $СНС = A_{2i}$ *ТО* $АНС_2 = B_i$

где $АНС_1$, $СНС$ – входные переменные, $АНС_2$ – выходные переменные, A_{1i}, A_{2i} и B_i – нечеткие множества, которые определены на универсальных множествах X_1, X_2, Y, соответственно.

В эксперименте рассмотрены четыре типа функций принадлежности: треугольные, трапециевидные, параболические и гауссовы, причем каждый из типов функций принадлежности декомпозирован на подтипы по степени нечеткости (значению базиса).

Примем в качестве эталонных значения, полученные в псевдофизической логике оценок величин, где абсолютная оценка выводится путем линейного сдвига, задаваемого оценкой сравнения [5].

Качество работы системы определяется следующими скалярными показателями: среднеквадратической ошибкой, средней абсолютной ошибкой, максимальной ошибкой.

Проведенные в работе исследования показали, что время и качество нечетких выводов зависят от вида функции принадлежности (треугольная, трапециевидная, параболическая, гауссова) и степени их нечеткости (значению базиса), способа задания *t*-оператора (Заде, вероятностные, Лукасевича, Швайцера-Скляра), способа задания самого нечеткого вывода (аппроксимация Мамдани, формальнологический), количества лингвистических термов во множествах АНС- и СНС-оценок.

Аппроксимация Мамдани.

Лучшими функциями принадлежности с точки зрения качества и времени вывода являются такие треугольные функции, графики которых для двух соседних лингвистических термов пересекаются на уровне 0.5. С ростом числа лингвистических термов увеличивается качество вывода, но время вывода растет в большей степени, чем качество вывода. Причем при использовании параметрической функции Швайцера-Скляра с ростом числа элементов выигрыш в качестве малозначителен, так, например, максимальные ошибки для пяти и семи элементных терм-множеств равны, соответственно, 0.01798 и 0.01603, а средние абсолютные 0.00490 и 0.00385, однако время вывода при этом возрастает почти в два раза. Лучшие выводы получены для дефаззификации методом центра тяжести. Возможность варьировать параметром в функции Швайцера-Скляра сделала ее лучшей с точки зрения качества вывода для функций принадлежности с небольшим и средним базисом, функция Лукасевича является лучшей для функций принадлежности с большим и очень большим базисом.

Абсолютно лучший результат получен при нарушении принципа двойственности (2), когда конъюнкция задана вероятностной функцией, а дизъюнкция – функцией Лукасевича. Так для пяти лингвистических термов и треугольных функций принадлежности, графики которых для двух соседних

термов пересекаются на уровне 0.5, абсолютная ошибка появляется только в пятом знаке после запятой, для девяти – в шестом [6].

Формальнологический подход.

Указанный способ нечеткого вывода в строгом смысле предполагает выполнение следующих условий:

1) соблюдение принципа двойственности при задании операций конъюнкции и дизъюнкции;

2) задание операции импликации посредством оператора *S*-типа;

3) совпадение типов *t*-конормальных функций, задающих операции импликации и дизъюнкции.

Формальнологический подход в нестрогом смысле предполагает выполнение только второго условия.

Качество нечеткого вывода при формальнологическом подходе в строгом и нестрогом смысле для подавляющего большинства способов задания логических операций и функций принадлежности уступает аппроксимации Мамдани. Качество вывода остается низким при увеличении количества лингвистических термов. Однако для формальнологического подхода в нестрогом смысле, когда конъюнкция задана функцией Лукасевича, а дизъюнкция – вероятностной функцией для треугольных функций принадлежности, графики которых для двух соседних термов пересекаются на уровне 0.5, эффективность вывода (время и качество) не уступает аппроксимации Мамдани.

Заключение.

Итак, формальнологический подход предполагает выражение импликации через *t*-конорму и соблюдение принципа двойственности *t*-норм и *t*-конорм при описании логических операций конъюнкции и дизъюнкции. Для получения эффективных результатов принцип двойственности не соблюдается. Таким образом, на поставленный в заголовке статьи вопрос ответ в общем случае – «нет» (формальнологический вывод в строгом смысле), но для конкретных сочетаний *t*-норм и *t*-конорм – ответ «да» (формальнологический вывод в нестрогом смысле). В общем случае качество вывода при формальнологическом подходе для подавляющего большинства способов задания логических операций и функций принадлежности значительно хуже, чем при аппроксимации Мамдани.

Список литературы

1. Kosko B. Fuzzy systems as universal approximators // IEEE Transactions on Computers – 1994. – v. 43. – p. 1329-1333.

2. Emami M. R., Turksen I. B., Goldenberg A. A. A unified parameterized formulation of reasoning in fuzzy modeling and control // Fuzzy Sets and Systems. – 1999. – v. 108. – p. 59-81.

3. Ходашинский И. А. Методы и модели оценки величин // Труды Международной научно-технической конференции «Интеллектуальные системы» (IEEE AIS`04) Т.1. – М.: Физматлит, 2004. – с. 141-146.

4. Ходашинский И. А. Нечеткологическое оценивание величин // Известия Томского политехнического университета. –2003. – т. 306, №3. – с. 10-15.

5. Ходашинский И. А. Псевдофизическая логика оценок величин // Известия АН СССР. Техническая кибернетика – №5, 1988. с. 96-107.

6. Ходашинский И. А. Формальнологический метод и аппроксимация Мамдани в нечетком оценивании величин // Автометрия. – 2006. –№ 1. – с. 55-67.

Статистические свойства оценки вариограммы гауссовского случайного процесса

Цеховая Т. В., *доцент, к.ф.-м.н.,*
Белорусский государственный университет, Беларусь, Минск, пр. Независимости,4,
E-mail: Tsekhavaya@bsu.by

Построена оценка вариограммы гауссовского случайного процесса с непрерывным временем. Найдены выражения для первых двух моментов исследуемой статистики. При условии, что ряд из вариограмм абсолютно сходится, исследовано асимптотическое поведение ковариации и дисперсии оценки вариограммы.

Ключевые слова: временной ряд, гауссовский случайный процесс, вариограмма, оценка.

В настоящее время проблема построения и изучения статистических свойств оценок основных характеристик временных рядов является достаточно актуальной. Интерес представляет статистический анализ оценок ковариационной функции и вариограммы, поскольку они являются основными мерами зависимости наблюдений за временными рядами. При этом часто ограничиваются исследованием только моментов первых двух порядков построенных оценок, так как они дают информацию об определенных свойствах изучаемых статистик. Так, например, в статье [1] рассматривался стационарный в широком смысле случайный процесс с дискретным временем. Найдены выражения для математического ожидания, ковариации, дисперсии оценок ковариационной функции и вариограммы через временные, а также частотные характеристики процесса. Исследованы асимптотические свойства моментов построенных статистик при ограничениях на ковариационную функцию и спектральную плотность рассматриваемого процесса.

При решении многих прикладных задач исследователи пытаются свести изучение исходных наблюдений к теории нормальных случайных процессов. Это связано с тем, что вышеуказанная теория разработана наиболее полно. Заметим также, что гауссовские случайные процессы принадлежат классу внутренне стационарных случайных процессов, которые адекватно описывают многие математические модели в геологии, экологии, эпидемиологии и т.д. В данной работе обобщаются вопросы исследования статистических свойств оценок вариограммы нормальных случайных процессов, рассмотренные в [2, 3].

Пусть $X(s)$, $s \in R$, – внутренне стационарный гауссовский случайный процесс с нулевым математическим ожиданием, дисперсией σ^2 и неизвестной вариограммой

$$2\gamma(h) = D\{X(s+h) - X(s)\},\ s,\ h \in R.$$

Нетрудно видеть, что величина

$$\{X(s+h) - X(s)\}^2 = 2\gamma(h) \cdot \chi_1^2,$$

где χ_1^2 – случайная величина, распределенная по закону хи-квадрат с одной степенью свободы. Следовательно,

$$M\{X(s+h) - X(s)\}^2 = 2\gamma(h),$$

$$D\{X(s+h) - X(s)\}^2 = 2\{2\gamma(h)\}^2.$$

Пусть $X(1)$, $X(2)$,..., $X(n)$ – n последовательных, полученных через равные промежутки времени наблюдений за процессом $X(s)$, $s \in R$. В качестве оценки вариограммы рассмотрим статистику вида

$$2\tilde{\gamma}(h) = \frac{1}{n-h}\sum_{s=1}^{n-h}(X(s+h) - X(s))^2, \qquad (1)$$

$h = \overline{0, n-1}$. Положим $\tilde{\gamma}(-h) = \tilde{\gamma}(h),\ \ h = \overline{0, n-1}$, и $\tilde{\gamma}(h) = 0$ для $|h| \geq n$.

Найдем выражения для первых двух моментов оценки (1).

Теорема 1. *Для оценки $2\tilde{\gamma}(h)$, заданной равенством (1), имеют место следующие соотношения:*

$$M\,2\tilde{\gamma}(h) = 2\gamma(h),$$

$$\operatorname{cov}\{2\tilde{\gamma}(h_1), 2\tilde{\gamma}(h_2)\} =$$

$$= \frac{2}{(n-h_1)(n-h_2)}\sum_{t=1}^{n-h_1}\sum_{s=1}^{n-h_2}\{\gamma(t+h_1-s) + \gamma(t-s-h_2) - \gamma(t+h_1-s-h_2) - \gamma(t-s)\}^2, \qquad (2)$$

$$\mathrm{D}\{2\tilde{\gamma}(h)\} = \frac{2}{(n-h)^2}\sum_{t,s=1}^{n-h}\{\gamma(t-s+h) + \gamma(t-s-h) - 2\gamma(t-s)\}^2, \qquad (3)$$

где $\gamma(h)$, $h \in R$, – семивариограмма процесса $X(s)$, $s \in R$, $h, h_1, h_2 = \overline{0, n-1}$.

Доказательство. Из определения вариограммы и свойств математического ожидания, первое утверждение теоремы вытекает очевидным образом.

Используя определение ковариации, подставляя вместо $2\tilde{\gamma}(h)$ ее выражение в явном виде, запишем

$$\operatorname{cov}\{2\tilde{\gamma}(h_1), 2\tilde{\gamma}(h_2)\} =$$

$$= \frac{1}{(n-h_1)(n-h_2)} \sum_{t=1}^{n-h_1} \sum_{s=1}^{n-h_2} \operatorname{cov}\{(X(t+h_1)-X(t))^2, (X(s+h_2)-X(s))^2\}.$$

Учитывая определение коэффициента корреляции, утверждения лемм 1 и 3 [3], получим

$$\operatorname{cov}\{2\tilde{\gamma}(h_1), 2\tilde{\gamma}(h_2)\} =$$

$$= \frac{2\{2\gamma(h_1)\}\{2\gamma(h_2)\}}{(n-h_1)(n-h_2)} \sum_{t=1}^{n-h_1} \sum_{s=1}^{n-h_2} \left\{ \frac{\gamma(t+h_1-s)+\gamma(t-s-h_2)-\gamma(t+h_1-s-h_2)-\gamma(t-s)}{\sqrt{2\gamma(h_1)}\sqrt{2\gamma(h_2)}} \right\}^2,$$

откуда следует требуемое равенство (2) для ковариации.

Соотношение (3) для дисперсии оценки вариограммы $2\tilde{\gamma}(h)$ нетрудно получить из (2), если положить $h_1 = h_2 = h$. Таким образом,

$$\mathrm{D}\{2\tilde{\gamma}(h)\} = \frac{2\{2\gamma(h)\}^2}{(n-h)^2} \sum_{t,s=1}^{n-h} \left\{ \frac{\gamma(t-s+h)+\gamma(t-s-h)-2\gamma(t-s)}{2\gamma(h)} \right\}^2 =$$

$$= \frac{2}{(n-h)^2} \sum_{t,s=1}^{n-h} \{\gamma(t-s+h)+\gamma(t-s-h)-2\gamma(t-s)\}^2 .$$

Теорема доказана.

Исследуем асимптотическое поведение моментов второго порядка построенной оценки $2\tilde{\gamma}(h)$, $h, = \overline{0, n-1}$.

Теорема 2. *Если имеет место соотношение*

$$\sum_{h=-\infty}^{+\infty} |\gamma(h)| < \infty, \qquad (4)$$

то

$$\lim_{n\to\infty} (n - \min\{h_1, h_2\}) \operatorname{cov}\{2\gamma(h_1), 2\gamma(h_2)\} = 2 \sum_{m=-\infty}^{+\infty} \{\gamma(m-h_2)+\gamma(m+h_1)-\gamma(m+h_1-h_2)-\gamma(m)\}^2, \qquad (5)$$

$$\lim_{n\to\infty} (n-h) D\{2\gamma(h)\} = 2\left[(2\gamma(h))^2 + 2\sum_{m=1}^{+\infty} \{\gamma(m-h)+\gamma(m+h)-2\gamma(m)\}^2\right], \qquad (6)$$

где $\gamma(h)$, $h \in R$, – семивариограмма процесса $X(s)$, $s \in R$, $h, h_1, h_2 = \overline{0, n-1}$.

Доказательство. Рассмотрим равенство (2). Пусть $h_1 > h_2$. Сделаем замену переменных: $t=t$, $t-s=m$, тогда

$$\operatorname{cov}\{2\tilde{\gamma}(h_1), 2\tilde{\gamma}(h_2)\} = \frac{2}{n-h_2} \left[\sum_{m=-(n-h_2-1)}^{n-h_1-1} \{\gamma(m+h_1)+\gamma(m-h_2)-\gamma(m+h_1-h_2)-\gamma(m)\}^2 - \right.$$

$$\left. - \frac{2}{n-h_1} \sum_{m=1}^{n-h_1-1} m\{\gamma(m+h_1)+\gamma(m-h_2)-\gamma(m+h_1-h_2)-\gamma(m)\}^2 \right].$$

Аналогично рассуждаем для случая $h_1 < h_2$. Объединяя полученные результаты, учитывая (4), вытекает требуемое предельное соотношение (5) для ковариации.

Равенство (6) для дисперсии оценки вариограммы $2\tilde{\gamma}(h)$ нетрудно получить из (5), если положить $h_1 = h_2 = h$.

Следствие. *Из теоремы 2 вытекает, что*

$$\lim_{n\to\infty} D\{2\tilde{\gamma}(h)\} = 0\,,\ h = \overline{0,n-1}.$$

В силу первого утверждения теоремы 1 и вышеуказанного следствия получаем, что $2\tilde{\gamma}(h)$ является состоятельной в среднеквадратическом смысле оценкой для вариограммы $2\gamma(h)$, $h \in R$.

Список литературы

1. Труш Н. Н., Цеховая Т. В. Исследование статистических свойств оценок вариограммы и ковариационной функции // Вести НАН Беларуси. Сер.1: физ.-мат. наук. – 2001. – №2.– С. 24-29.

2. Cressie N. Fitting variogram models by weighted least squares // Jour. Inter. Assoc. Math. Geol.– 1985. – Vol. 17, № 5.– P. 563-586.

3. Цеховая Т. В. Первые два момента оценки вариограммы гауссовского случайного процесса // Межд. мат. конференция "Дифференциальные уравнения и системы компьютерной алгебры" (DE&CAS' 2005), 5-8 октября, 2005г. / Брест, 2005.– С. 78-82.

О сложности приближения непрерывных функций детерминированными функциями с задержкой

Черепов А. Н.,
доцент Смоленского филиала МЭИ(ТУ)
214036,г. Смоленск, ул. Попова, д. 98, кв. 129,
ancherepov@mail.ru

А. Н. Колмогоровым была поставлена задача приближения непрерывных функций детерминированными [1,2]. Поскольку для заданного ε не для всякой непрерывной функции существует ε - равная ей детерминированная функция, то были предприняты попытки рассмотрения приближений непрерывных функций функциями, близкими к детерминированным. В.А.Буевич предложил исследовать возможность приближения непрерывных функций детерминированными функциями с задержкой. Некоторые результаты, связанные с возможностью приближения непрерывных функций детерминированными функциями с задержкой, были получены учениками В.А.Буевича [3 – 7]. В предлагаемой работе содержится обобщения этих результатов и некоторые новые утверждения о сложности приближения непрерывных функций.

Рассмотрим множество всех бесконечных двоичных последовательностей E. Множество всех функций вида $f : E^n \to E$ будем обозначать P. Предположим, что a_1 , a_2 , …, a_n произвольные последовательности из E, тогда $\tilde{a} = (a_1 , a_2 , \ldots, a_n)$ - набор таких последовательностей. Пусть $a_1 \mid k$, $a_2 \mid k$, …, $a_n \mid k$ первые k членов последовательностей $a_1, a_2, \ldots, a_n$ соответственно, тогда $\tilde{a}|k = (a_1| k, a_2| k, \ldots, a_n| k)$.

Назовем функцию f детерминированной, если для всех $k=1,2,\ldots$ выполнено: $\forall\ \tilde{a}\ ,\tilde{b}:\ \tilde{a} \mid k = \tilde{b} \mid k \ \Rightarrow\ f(\tilde{a})| k = f(\tilde{b})| k$.

Класс всех детерминированных функций обозначим $P_Д$, все остальные функции из P будем называть недетерминированными.

Определение 1. Говорим, что функция f является детерминированной функцией с задержкой τ, где τ − произвольное неотрицательное целое число, если для любого $i=1,2,3,\ldots$ и любых $\tilde{a}$, $\tilde{b}$ выполнено $\tilde{a} \mid i+\tau = \tilde{b} \mid i+\tau \ \Rightarrow\ f(\tilde{a}) \mid i = f(\tilde{b}) \mid i$.

Множество всех детерминированных функций с задержкой τ обозначим $P_Д^\tau$. Заметим, что $P_Д^0 = P_Д$. В класс $P_Д^\infty$ включим все функции множеств $P_Д^\tau$ при $\tau=0, 1, \ldots$

Множество функций $P_Д^\tau$ можно определить и следующим образом.

Определение 2. Говорим, что функция f является детерминированной функцией с задержкой τ, где τ − произвольное неотрицательное целое число, если существует такая детерминированная функция g, что для любого $\tilde{a}$ и $b = g(\tilde{a})$, $b = b(1)b(2)b(3)\ldots$ значение функции f на $\tilde{a}$ равно $f(\tilde{a}) = b(\tau+1)b(\tau+2)b(\tau+3)\ldots$

Утверждение 1. *Определения 1 и 2 эквивалентны.*

Из определения 2 следует, что функцию n аргументов из множества $P_Д^\tau$ можно трактовать так: берется дискретное детерминированное устройство с n входами, преобразующее бесконечные входные последовательности из нулей и единиц в такую же выходную последовательность и рассматривается выход этого устройства не с первого момента времени, а с момента времени $\tau + 1$. Осуществляемое преобразование и считается функцией множества $P_Д^\tau$. Таким образом, функции с задержкой являются естественным обобщением детерминированных функций.

Задача приближения непрерывных функций функциями специального вида очень часто появляется в математике. Детерминированные функции и детерминированные функции с задержкой являются хорошей моделью реальных устройств – преобразователей сигналов, работающих в дискретные моменты времени. Переход от дискретных функций к функциям вещественных переменных, позволяет ставить вопрос о приближении непрерывных функций функциями с задержкой.

Сопоставим каждой двоичной последовательности $a(1)a(2)\ldots a(i)\ldots$ некоторое число отрезка $[0,1]$ равное $0, a(1)a(2)\ldots a(i)\ldots$. Из двух возможных представлений $0,a(1)a(2)\ldots a(i)100\ldots=0, a(1)a(2)\ldots a(i)0111\ldots$ выберем первое.

Определение 3. Пусть $\alpha = 0,\alpha_1\alpha_2\alpha_3\ldots, \beta = 0,\beta_1\beta_2\beta_3\ldots$ действительные числа отрезка $[0,1]$ и k – натуральное число. Тогда будем считать, что $\alpha \mid k = \beta \mid k$, если у чисел α, β совпадают первые k двоичных разряда. Для точек $\tilde{a}, \tilde{b} \in [0,1]^n$ имеем, что $\tilde{a} \mid k = \tilde{b} \mid k$, если в каждой координате выполнено равенство $a_i \mid k = b_i \mid k$. Действительную функцию f назовем детерминированной, если для любого $k=1, 2, 3\ldots$ из $\tilde{\alpha} \mid k = \tilde{\beta} \mid k$ следует, что $f(\tilde{\alpha}) \mid k = f(\tilde{\beta}) \mid k$. Класс всех детерминированных функций обозначим $D_Д$. В множество $D_Д^\tau$ действительных детерминированных функций с задержкой τ включим все функции, удовлетворяющие свойству:

для любого $i=1, 2, 3\ldots$ из $\tilde{a} \mid i+\tau = \tilde{b} \mid i+\tau$ следует $f(\tilde{a}) \mid i = f(\tilde{b}) \mid i$.

Класс $D_Д^\infty$ определим как объединение всех множеств $D_Д^\tau$. Множества $D_Д^\tau$ и $D_Д^\infty$ могут быть построены по соответствующим им классам дискретных функций, при потере части информации о значениях дискретных функций на последовательностях вида $a(1)a(2)\ldots a(i)0111\ldots$

Определение 4. Пусть $\varepsilon > 0$, будем говорить, что функция $d(\tilde{x})$ ε - равна функции $f(\tilde{x})$ на единичном кубе $[0,1]^n$, если при любом $\tilde{x} \in [0,1]^n$ имеем, что $\left|f(\tilde{x}) - d(\tilde{x})\right| < \varepsilon$. Будем также говорить, что $d(\tilde{x})$ ε – приближает $f(\tilde{x})$.

Пусть $C[0,1]^n$ - это множество непрерывных на единичном кубе $[0,1]^n$ функций, принимающих значения из отрезка $[0,1]$. Для любых точек $\tilde{x}, \tilde{y} \in [0,1]^n$ будем считать, что $\left| \tilde{x} - \tilde{y} \right| = max\left|x_i - y_i\right|$, где максимум берется по всем координатам.

Оказывается, что не всякую функцию множества $C[0,1]^n$ можно ε – приблизить детерминированными функциями класса $D_Д$. Но функций класса $D_Д^\infty$ уже достаточно для

решения этой задачи. В работе [4] следующее утверждение было доказано для функций несколько более широкого чем $D^{\infty}_{Д}$ класса.

Теорема. *Для любого* $\varepsilon > 0$ *и любой функции* $f(\tilde{x}) \in C[0,1]^n$ *существует число* $\tau \geq 0$ *и функция* $d(\tilde{x}) \in D^{\tau}_{Д}$ *такие, что* $d(\tilde{x})$ ε *- равна* $f(\tilde{x})$.

Определение 5. Для любой непрерывной функции f и любого ε назовем сложностью реализации функции f, наименьшее τ такое, что существует функция $d(\tilde{x})$ класса $D^{\tau}_{Д}$ ε - равная f. Сложность реализации функции f будем обозначать $L(f, \varepsilon)$, а при $\varepsilon = 2^{-k}$ будем считать, что $L(f, \varepsilon) = L(f, k)$.

Приведем примеры ε - приближений некоторых непрерывных функций функциями из множества $D^{\tau}_{Д}$, стараясь сделать задержку τ минимальной.

Функция сложения *x + y*. Для того чтобы эта функция принимала значения от 0 до 1, рассмотрим ее некоторое ограничение. Будем считать, что $x+y=min(x+y,1)$.

Утверждение 2. *Для функции* $x+y$ *при любом* $\varepsilon = 2^{-k}$

$$L(x+y, k)=k-1.$$

Следствие. *Функцию сложения нельзя приблизить с точностью* $\varepsilon = 2^{-k}$ *функциями класса* $D_{Д}$ *при любом целом* $k>1$.

Функция умножения *x•y*, где 0≤x, y≤1.

Утверждение 3. *Для функции* $x \cdot y$ *при любом* $\varepsilon = 2^{-k}$

$$L(x \cdot y , k)=k-1.$$

Следствие. *Функцию умножения нельзя приблизить с точностью* $\varepsilon = 2^{-k}$ *функциями класса* $D_{Д}$ *при* $k>1$.

Функция $\sqrt[n]{x}$, где *n* натуральное число.

Утверждение 4. *Для функции* $\sqrt[n]{x}$ *при любом* $\varepsilon = 2^{-k}$

$$L(\sqrt[n]{x} , k)=k(n-1).$$

Следствие. *Функцию* $\sqrt[n]{x}$ *нельзя приблизить с точностью* $\varepsilon = 2^{-k}$ *функциями класса* $D_{Д}$ *при* $k>1$.

Автор выражают свою признательность В. А. Буевичу за постановку задачи и существенную помощь, оказанную при окончательном оформлении результатов.

Список литературы

1. Офман Ю. Об алгоритмической сложности дискретных функций //Доклады АН СССР.- 1962. -Т. 145, № 1. – С.48-51.

2. Офман Ю. О приближенной реализации непрерывных функций на автоматах// Доклады АН СССР.-1963. -Т. 152,№ 4. –С. 823-826.

3. Тюленев Н. Ф. Приближение непрерывных функций дискретными // Сб. трудов семинара по дискретной математике и ее приложениям. – М.: Изд-во мех.-мата МГУ, - 1997. –С.148 – 151.

4. Тюленев Н. Ф. О приближении непрерывных функций дискретными //Конструкции в алгебре и логике. – Тверь, 1990. С.110-116.

5. Черепов А. Н., Черепов И. А. О представлении недетерминированных функций детерминированными // Тезисы докладов XIII международной конференции «Проблемы теоретической кибернетики». Часть 2. – М.: 2002. – С. 191.

6. Черепов А. Н., Черепов И. А. О классификации недетерминированных функций // Сб. трудов семинара по дискретной математике и ее приложениям. – М.: Изд-во мех.-мата МГУ, - 2004.

7. Черепов И. А. О приближении непрерывных функций детерминированными функциями с задержкой // Сб. трудов семинара по дискретной математике и ее приложениям. – М.: Изд-во мех.-мата МГУ, - 2004.

Множества, оценивающие параметр формы сигнала[2]

Чуличков А. И.,
профессор физического факультета
МГУ им. М.В.Ломоносова,
e-mail: ach@cmp.phys.msu.su

На основе теории измерительно-вычислительных систем [1-2] и морфологических методов анализа сигналов [3] получены оценки параметров формы сигнала в виде оценивающих множеств. Форма определена как инвариант преобразований сигнала, моделирующих изменение условий его регистрации.

Введение.

При анализе кусочно-непрерывных сигналов часто наиболее интересным является положение точек экстремума, и совершенно не важно, как изменяются значения сигналов на интервалах между экстремумами. Информация о сцене, заключенная в ее изображении, тоже не сводится к абсолютным значениям яркости каждой точки поля зрения, более важны размеры и форма характерных особенностей распределения яркости по полю зрения. Для формализации такого рода информации в морфологическом анализе [3] определяется операция сравнения сигналов по форме: сигнал рассматривается как функция $f(\cdot)$, заданная на некотором множестве X и принимающая значение в евклидовом пространстве R^k, $1 \le k < \infty$; вводится измеримое пространство (L, A) всех сигналов R с заданной на нем мерой. Пусть F - класс преобразований $F : R^k \to R^k$. Сигнал $g(\cdot) \in L$ по форме не сложнее, чем $f(\cdot) \in L$, если $g(x) = F(f(x))$ почти всюду на X. Например, если $f(\cdot)$ –кусочно-непрерывная борелевская функция, заданная на отрезке $[0,1] \subset R^1$ и принимающая числовые значения, и F - класс монотонно неубывающих преобразований, то точки экстремума сигнала $F * f \in L$ являются и точками экстремума сигнала f, если $F * f(x) = F(f(x))$ почти всюду на X. В то же время интервалы строгой монотонности функции $f(\cdot)$ могут оказаться множествами постоянства функции $F * f(\cdot)$, и «характер зависимости» («*форма*») $F * f(x)$ от $x \in X$ будет «проще», чем $f(x)$, $x \in X$. Множество всех сигналов, форма

[2] Работа выполнена при финансовой поддержке РФФИ, грант № 05-01-00615-а.

которых не сложнее, чем $f(\cdot) \in L$, в морфологическом анализе [3] носит название формы сигнала $f(\cdot)$. В рассматриваемом случае форма $V_f = \{g \in L : g = F * f, F \in \mathsf{F}\}$ сигнала f является конусом в L.

Пусть задан параметрический класс форм сигналов $\{V_\lambda,\ \lambda \in \Lambda\}$, предъявляется сигнал, искаженный аддитивной помехой, и требуется оценить значение параметра его формы. Классические методы морфологического анализа в качестве оценки дают значение λ_0 параметра той формы V_{λ_0}, к которой наиболее близок предъявленный сигнал. Интерес, однако, представляют оценки, оптимальные по точности. В данной работе построены оценки параметров формы в виде оценивающих множеств, размер которых выбирается из соображений оптимальности.

Оценка параметра формы при известной дисперсии погрешности измерений.

Пусть $L = R^n$ - конечномерное евклидово пространство векторов, заданных своими координатами $f = (f_1,..., f_n) \in R^n$. Рассмотрим параметрическое семейство выпуклых замкнутых множеств $\{V_\lambda,\ \lambda \in \Lambda\}$, представляющих собой класс возможных форм исследуемого сигнала f; наблюдение сигнала $f \in V_\lambda$ производится по схеме

$$\xi = f + \nu, \tag{1}$$

где значение параметра формы λ сигнала $f(\cdot) \in R^n$ неизвестно, $\lambda \in \Lambda$, а погрешность измерения $\nu \in R^n$ - нормально распределенный вектор с нулевым математическим ожиданием $\mathbf{E}\nu = 0$ и некоррелированными координатами, дисперсии которых равны σ^2: $\nu \sim N(0,\sigma^2 I)$. Требуется по предъявленному сигналу $\xi \in R^n$ оценить значение параметра его формы $\lambda \in \Lambda$.

Построим множество $I_p(\xi) \subset \Lambda$, оценивающее параметр формы, из следующих соображений. Значение $\lambda_0 \in \Lambda$ принадлежит множеству $I_p(\xi)$, если результат измерения (1) может быть представлен в виде суммы некоторого элемента $f \in V_{\lambda_0}$ и реализации случайного вектора $\nu \sim N(0,\sigma^2 I)$. Так как $\xi \sim N(f,\sigma^2 I)$, то при $f \in V_{\lambda_0}$ реализация случайного вектора ξ отстоит от V_{λ_0} на расстояние, не большее $\|\nu\|$. При этом чем больше это расстояние, тем менее возможна такая реализация ξ при $f \in V_{\lambda_0}$. Следуя [1-3], будем считать, что мерой согласия реализации ξ с гипотезой $\xi \sim N(f,\sigma^2 I)$, $f \in V_{\lambda_0}$, является надежность $\alpha_{\lambda_0}(\xi)$ этой гипотезы при альтернативе $f \notin V_{\lambda_0}$, которую определим как вероятность $\alpha_{\lambda_0}(\xi) = P\left(\|\eta - P_{\lambda_0}\eta\|^2 \ge \|\xi - P_{\lambda_0}\xi\|^2\right)$, где $\eta \sim N(\mu,\sigma^2 I)$, $P_{\lambda_0}\xi$ - проекция ξ на V_{λ_0}, а $\mu \in V_{\lambda_0}$ - наиболее близкая к ξ точка множества V_{λ_0}, т.е. $\mu = P_{\lambda_0}\xi$. Иными словами, надежностью рассматриваемой гипотезы $f \in V_{\lambda_0}$ при альтернативе $f \notin V_{\lambda_0}$ является вероятность получить в эксперименте (5) результат, согласующийся с гипотезой так же, как ξ, или еще хуже. Оценка надежности $\alpha_{\lambda_0}(\xi)$ методом Монте-Карло равна отношению числа случаев выполнения неравенства $\|\eta_k - P_{\lambda_0}\eta_k\|^2 \ge \|\xi - P_{\lambda_0}\xi\|^2$, где η_k - k-я реализация случайного вектора $\eta \sim N(P_{\lambda_0}\xi,\sigma^2 I)$, к общему числу реализаций (при этом математическое ожидание $\mu = P_{\lambda_0}\xi$ вектора η считается фиксированным).

Множество, оценивающее значение параметра формы λ по результату измерения (1), определим следующим образом:

$$I_p(\xi) = \{\lambda \in \Lambda : \alpha_\lambda(\xi) \ge 1 - p\}. \tag{2}$$

Иными словами, в оценивающее множество попадают те и только те значения $\lambda \in \Lambda$, для которых гипотеза $f \in V_\lambda$ достаточно хорошо согласуется с результатом измерения (1). Параметр p в этом случае является оценкой снизу вероятности $P(\lambda_0 \in I_p(\xi) \big| f \in V_{\lambda_0})$ включения истинного значения λ_0 параметра формы в оценивающее множество: чем меньше пороговые значения в (2), тем больше вероятность $P(\lambda_0 \in I_p(\xi) \big| f \in V_{\lambda_0})$.

Оценка параметра формы при известной дисперсии погрешности измерений.

В этом случае распределение статистики $\|\xi - P_\lambda \xi\|^2$ зависит от неизвестной дисперсии σ^2 шума ν в (1), и поэтому не может служить для количественной характеристикой включения $f \in V_\lambda$. Для этой цели лучше подходит статистика [3]

$$\tau_\lambda(\xi) = \frac{\|\xi - P_\lambda \xi\|^2}{\|P_\lambda \xi - P_0 \xi\|^2}, \tag{3}$$

где P_0 – проектор на множество $V_0 = \{\mu = (\mu_1, ..., \mu_n) \in R^n : \mu_i = c, i = 1, ..., n, c \in (-\infty, \infty)\}$. Если V_λ – выпуклый замкнутый конус, для каждого $\lambda \in \Lambda$ содержащий V_0 и такой, что $P_\lambda(g + \mu) = P_\lambda g + \mu$ для всех $\mu \in V_0$ и $\lambda \in \Lambda$, то распределение статистики (3) не изменяется при умножении ξ на любое число, отличное от нуля, и при сложении ξ с любым вектором $\mu \in V_0$ (примером таких конусов является множество $V = \{f = (f_1, f_2, ..., f_n) : f_i \in (-\infty, \infty),\ i = 1, ..., n,\ f_1 \leqq f_2 \leqq ... \leqq f_n\}$, где «$\leqq$» означает либо «$\leq$», либо «$\geq$»; различным значениям параметра $\lambda \in \Lambda$ отвечает разный порядок следования этих знаков). Знаменатель дроби (3) дает отличие формы сигнала $P_\lambda \xi$ от формы сигнала, равного константе. Так как $P_0 \xi \in V_\lambda$ для любого $\lambda \in \Lambda$, то $P_0 \xi$ не несет информации о значении параметра λ. Разность $P_\lambda \xi - P_0 \xi \in V_\lambda$ есть «часть» сигнала ξ, лежащая в V_λ, а $\xi - P_\lambda \xi$ – «часть» сигнала ξ, не принадлежащая V_λ, и если $\tilde{\lambda}$ – истинное значение параметра формы, то $\|\xi - P_{\tilde{\lambda}} \xi\|^2 \leq \|\nu\|^2$, и $\|\xi - P_{\tilde{\lambda}} \xi\|^2$ может служить характеристикой величины погрешности. Если для некоторого $\lambda \in \Lambda$ $\|\xi - P_\lambda \xi\|^2 << \|P_\lambda \xi - P_0 \xi\|^2$, то естественно считать, что ξ есть результат наблюдения сигнала $f \in V_\lambda$ по схеме (1), а величина $\tau_\lambda(\xi)$, определенная в (3), характеризует согласие предположения $f \in V_\lambda$ с результатом измерения ξ. Значение $\tau_\lambda(\xi)$ тем больше, чем ближе сигнал ξ к константе по сравнению с близостью ξ к V_λ. Однако с формальной точки зрения, величина $\tau_\lambda(\xi)$ тем меньше, чем более правдоподобно сделанное на основании результата измерения ξ утверждение о том, что верна гипотеза

$$H_\lambda: \ \xi \sim N(a, \sigma^2 I), \qquad a \in V_\lambda \setminus V_0 \tag{4}$$

при альтернативе

$$K: \ \xi \sim N(a, \sigma^2 I), \qquad a \in V_0. \tag{5}$$

Морфологический критерий проверки гипотезы (4) при альтернативе (5) определяется критическим множеством [1]:

$$S_\lambda = \{z \in R^n : \tau_\lambda(\xi) \geq \delta\}. \tag{6}$$

Если $\xi \notin S_\lambda$, то гипотеза (4) принимается, и можно считать, что форма сигнала ξ достаточно близка к форме V_λ. Следуя [1,2], охарактеризуем согласие гипотезы с экспериментом минимальным уровнем критерия, отвергающим гипотезу (4) в пользу (5) по наблюдению ξ. Эта характеристика, называемая надежностью гипотезы (4), в данном случае равна

$$\alpha_{H\lambda}(\xi) = \sup\{P(\tau_\lambda(\zeta) \geq \tau_\lambda(\xi)) \mid \zeta \sim N(\mu, \sigma^2 I),\ \mu \in V_\lambda \setminus V_0,\ \sigma^2 > 0\}.$$

Если вероятность ошибки первого рода (уровень критерия) возрастает при стремлении гипотезы к множеству альтернатив, то вероятность $P(\tau_\lambda(\zeta) \geq \tau_\lambda(\xi))$ при $\zeta \sim N(\mu, \sigma^2 I),\ \mu \in V_0, \sigma^2 > 0$ может служить оценкой надежности $\alpha_{H\lambda}(\xi)$. Она не зависит от параметров $\mu \in \overline{V_0}$ и $\sigma^2 > 0$ и может быть вычислена методом Монте-Карло путем разыгрывания реализаций вектора $\zeta \sim N(0, I)$, и подсчета частоты реализаций, для которых $\tau_\lambda(\zeta) < \tau_\lambda(\xi)$. Так же, как и в предыдущем пункте, этой оценке можно придать смысл вероятности получить результат измерения (1), согласующийся с гипотезой так же, как ξ, или еще хуже.

Заметим, что если V_λ - k-мерное подпространство R^n, то случайная величина

$$\frac{n-k}{\tau_\lambda(\zeta)(k-1)}$$

при $\zeta \sim N(\mu, \sigma^2 I)$, $\mu \in V_0$, контролируется распределением Снедекора-Фишера с $k-1, n-k$ степенями свободы, а область S_λ в (6) определяет равномерно наиболее мощный критерий проверки гипотезы (4) при альтернативе (5) в классе критериев, инвариантных к преобразованиям, определяемых симметрией задачи проверки гипотез (4), (5) [4,5].

Множество $I_p(\xi)$, оценивающее параметр $\lambda \in \Lambda$ с вероятностью не меньшей p построим по следующему правилу: будем считать, что $I_p(\xi)$ состоит из тех и только тех значений $\lambda \in \Lambda$, для которых надежность альтернативы (5) при гипотезе (4) не меньше p: $\alpha_\lambda(\xi) = P(\tau_\lambda(\zeta) \geq \tau_\lambda(\xi)) \geq p$, что влечет неравенство $\tau_\lambda(\xi) \leq \delta(\varepsilon)$.

Чем меньше $\delta(\varepsilon)$, тем меньше (по включению) оценивающее множество $I_p(\xi)$, и тем точнее локализуется оцениваемое значение $\lambda \in \Lambda$.

Список литературы

[1] Пытьев Ю. П. Методы математического моделирования измерительно-вычислительных систем // М.:Физматлит, 2002. – 384с.

[2] Чуличков А. И. Основы теории измерительно-вычислитьельных систем. Стохастические линейные измерительно-вычислительные системы. // Тамбов: Изд-во Тамбовского гос. тех. ун-та. 2000. – 140с.

[3] Пытьев Ю. П. Задачи морфологического анализа изображений. – В сб.: Математические методы исследования природных ресурсов Земли из Космоса. М.:Наука. 1984г.

[4] Богданов И. В., Чуличков А. И. Применение локального морфологического фильтра при анализе изображений. - 6-я международная конференция «Распознавание образов и анализ изображений: новые информационные тенхнологии» РОАИ – 6 – 2002. В.Новгород, 2002, с. 71-74.

[5] Леман Э. Проверка статистических гипотез. – М.:Наука. 1979.

Энтропия и информация нечетких текстов [1]

Л. А. Шоломов,
Институт системного анализа РАН,
вед. научн. сотр.
117312 Москва, просп. 60-летия Октября, 9
E-mail: sholomov@isa.ru

Рассматриваются последовательности нечетких символов, возникающие в результате нечеткой записи исходных четких последовательностей. Исследуется задача оценки среднего размера двоичной информации, необходимого для восстановления исходных четких данных по наблюдаемым нечетким. Она формализована в терминах искажения информации в нечетких каналах. При некоторых статистических допущениях с использованием метода максимального правдоподобия получено решение этой задачи для различных типов нечетких каналов. Описан итеративный алгоритм решения уравнений максимального правдоподобия, которые в общем виде не решаются, и доказана его сходимость.

Ключевые слова: нечеткие символы; нечеткий канал; частичные данные; энтропия; условная энтропия; информация; метод максимального правдоподобия

Нечеткие последовательности

С нечеткими и частично определенными данными имеют дело в задачах распознавания образов, управления, принятия решений, логического синтеза, генетики. Это указывает на целесообразность расмотрения нечетких данных в качестве самостоятельного объекта. Под нечеткими данными будем понимать последовательности символов, среди которых встречаются нечеткие. Если в нечетко написанной последовательности цифр некоторый символ может быть воспринят как 3 и как 5, его будем обозначать $a_{3,5}$, а в общем случае для нечеткого символа будем использовать обозначение a_T, где T — множество четких символов, в результате нечеткой записи которых он мог возникнуть. Нечеткими можно считать и частично определенные последовательности. Если, например, имеется частично определенная двоичная последовательность, составленная из 0, 1 и неопределенного символа $*$, то $*$ можно трактовать как нечеткий символ $a_{0,1}$. В [1] (краткое изложение имеется в [2]) изучалась энтропия нечетких данных и задача их сжатия. В настоящей работе рассматривается вопрос о мере информации, которую несет заданная последовательность нечетких символов о исходной (неизвестной нам) четкой последовательности, из которой она образовалась в результате нечеткой записи.

Введем соответствующие понятия. Задан алфавит $A_0 = \{a_0, a_1, \ldots, a_{m-1}\}$ *четких* символов. Положим $M = \{0, 1, \ldots, m-1\}$. Пусть некоторым непустым подмножествам $T \subseteq M$ сопоставлены символы a_T, которые будем называть *нечеткими*. Алфавит нечетких символов обозначим через A. Будем считать, что $A \supseteq A_0$ (символы a_i соответствуют одноэлементным множествам $T = \{i\}$). *Доопределением* символа $a_T \in A$ назовем всякий четкий символ a_i, $i \in T$, а доопределением последовательности в алфавите A — любую последовательность в алфавите A_0, полученную из исходной заменой всех ее символов некоторыми доопределениями.

Формализуем способ возникновения нечетких последовательностей из четких. Будем считать, что имеется *нечеткий канал* (без памяти), на вход которого подаются четкие символы, а с выхода снимаются нечеткие. Он характеризуется набором переходных вероятностей $p(T|i) = p(a_T|a_i)$, $a_i \in A_0$, $a_T \in A$, $p(T|i) = 0$ для $i \notin T$. Простейшим случаем такого канала является двоичный симметричный канал со стиранием [3,4], в котором четкие символы 0 и 1 с вероятностью ε переходят в неопределенный символ $*$ (стираются) и с вероятностью $1-\varepsilon$ сохраняются. Дальше будем полагать, что для любого четкого символа a_i имеется положительная вероятность $p(i|i) > 0$ того, что он не будет искажен, где $p(i|i)$ означает $p(\{i\}|i)$.

Пусть на выходе нечеткого канала наблюдается последоваиельность $\mathbf{a} = a_{T_1} \ldots a_{T_n}$ длины n. Обозначим через n_T число появлений в ней символа a_T, $\sum_T n_T = n$, а через $p_T = n_T/n$ — его

[1] Работа выполнена при поддержке Отделения информационных технологий и вычислительных систем РАН по программе фундаментальных исследований.

частоту. Положим $P = (p_T,\ a_T \in A)$. Будем считать, что в $\mathbf{a}$ присутствуют все четкие символы, т. е. $p_i > 0$, $i \in M$.

Будем полагать, что последовательность $\mathbf{a}$ возникла из некоторой четкой последовательности, символы a_i которой подавались на вход канала независимо с некоторыми неизвестными нам вероятностями q_i. Задача состоит в том, чтобы оценить информацию в нечеткой последовательности $\mathbf{a}$ о неизвестной исходной четкой последовательности.

Уравнения максимального правдоподобия

Для оценки параметров распределения $Q = (q_0, \dots, q_{k-1})$ на входе канала используем метод максимального правдоподобия. Вероятность символа a_T на выходе канала составляет $\sum_{i \in T} q_i p(T|i)$, а вероятность последовательности $\mathbf{a}$ равна

$$p_{\mathbf{a}}(Q) = \prod_T (\sum_{i \in T} q_i p(T|i))^{n_T}.$$

Распределение Q находится из условия максимизации $p_{\mathbf{a}}(Q)$, или, что эквивалентно, — минимизации величины

$$-\frac{\log p_{\mathbf{a}}(Q)}{n} = -\sum_T p_T \log \sum_{i \in T} q_i p(T|i) = \mathcal{H}(P, Q) \tag{1}$$

(здесь и дальше логарифмы двоичные). Положим $\mathcal{H}(P) = \min_Q \mathcal{H}(P, Q)$. Нетрудно убедиться, что для четкого канала (т. е. при $p(i|i) = 1$ и $p(T|i) = 0$ для $T \neq \{i\}$) величина $\mathcal{H}(P)$ совпадает с энтропией Шеннона $-\sum_i p_i \log p_i$.

Теорема 1 *Набор вероятностей Q минимизирует функцию $\mathcal{H}(P, Q)$ тогда и только тогда, когда при каждом i, $i \in M$, имеет место равенство*

$$\sum_{T:\, i \in T} \frac{p_T p(T|i)}{\sum\limits_{j \in T} q_j p(T|j)} = 1. \tag{2}$$

Минимизирующий набор Q единствен и все его компоненты положительны.

Доказательство. В силу сделанного допущения $p_i > 0$ выражение для $\mathcal{H}(P, Q)$ содержит слагаемые $-p_i \log(q_i p(i|i))$ для всех i. Поэтому компоненты q_i минимизирующего набора Q положительны.

Вогнутая по Q функция $-\mathcal{H}(P, Q)$ удовлетворяет условиям теоремы 4.4.1 из [3]. В соответствии с ней необходимым и достаточным условием того, что в точке Q достигается максимум функции $-\mathcal{H}(P, Q)$, является существование λ, при котором $-\partial\mathcal{H}(P, Q)/\partial q_i \leq \lambda$, $i \in M$, где строгие неравенства могут соответствовать лишь нулевым значениям q_i. В рассматриваемом случае с учетом положительности набора Q эти соотношения приобретают вид равенств

$$\log e \sum_{T \ni i} \frac{p_T p(T|i)}{\sum\limits_{j \in T} q_j p(T|j)} = \lambda, \quad i \in M. \tag{3}$$

Домножив их на q_i и просуммировав по $i \in M$, получаем в левой части

$$\log e \sum_{i \in M} \sum_{T \ni i} \frac{p_T q_i p(T|i)}{\sum\limits_{j \in T} q_j p(T|j)} = \log e \sum_T p_T \frac{\sum\limits_{i \in T} q_i p(T|i)}{\sum\limits_{j \in T} q_j p(T|j)} = \log e \sum_T p_T = \log e. \tag{4}$$

Правая часть дает $\lambda \sum_i q_i = \lambda$. Сравнивая, находим $\lambda = \log e$. Подставив это значение в (3), приходим к (2).

Осталось установить единственность минимизирующего набора Q. Предположим, что это не так и Q, Q' — два различных набора, на которых достигается значение $\mathcal{H}(P)$. Взяв произвольное μ, $0 < \mu < 1$, и применив неравенство Иенсена к выпуклой по Q функции $\mathcal{H}(P, Q)$, получаем

$$\mathcal{H}(P, \mu Q + (1-\mu)Q') \leq \mu\mathcal{H}(P, Q) + (1-\mu)\mathcal{H}(P, Q') = \mathcal{H}(P).$$

Кроме того, если наборы Q и Q' различаются в компоненте i, то

$$-p_i \log((\mu q_i + (1-\mu)q'_i)p(i|i)) < -\mu p_i \log(q_i p(i|i)) - (1-\mu)p_i \log(q'_i p(i|i))$$

и предыдущее неравенство является строгим. Это противоречит минимальности значения $\mathcal{H}(P)$. Терема доказана.

Единственный набор Q, минимизирующий функцию $\mathcal{H}(P,Q)$ (и максимизирующий вероятность $p_{\mathbf{a}}(Q)$), обозначим через $\hat{Q}$. Таким образом, $\mathcal{H}(P) = \mathcal{H}(P,\hat{Q})$. Набор $\hat{Q}$ однозначно находится из уравнений (2), Будем называть их *уравнениями максимального правдоподобия*, а $\hat{Q} = (\hat{q}_0, \ldots, \hat{q}_{m-1})$ — *распределением максимального правдоподобия*. Алгоритм вычисления $\hat{Q}$ будет приведен в заключительной части работы.

Условная энтропия и информация

Пусть Q — распределение на входе канала, P — наблюдаемое распределение (набор частот) на выходе. Теоретико-информационной характеристикой среднего размера двоичной информации на символ, позволяющей по выходной последовательности восстановить входную, является условная энтропия [3]

$$\mathcal{H}(Q|P) = -\sum_T p_T \sum_{i\in T} p(i|T) \log p(i|T). \tag{5}$$

Для ее вычисления необходимо знать условные вероятности $p(i|T) = p(a_i|a_T)$, $i \in T$.

При заданном распределении на входе Q и заданных переходных вероятностях $p(T|i)$ канала условные вероятности $p(i|T)$ могут быть найдены по формуле Байеса

$$p(i|T) = \frac{q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)}. \tag{6}$$

Скажем, что условные вероятности в (5) *согласованы с каналом*, если они совпадают с вычисленными по формуле (6). При известном распределении P на выходе набор условных вероятностей, согласованных с каналом, существует не при всяком Q, поскольку $p(a_i a_T) = p_T p(i|T)$ и, следовательно, должны быть выполнены соотношения

$$\sum_{T\ni i} p_T p(i|T) = q_i, \quad i \in M. \tag{7}$$

Теорема 2 *Набор условных вероятностей, согласованных с каналом, для входного распределения Q существует тогда и только тогда, когда оно является распределением максимального правдоподобия.*

Доказательство. С учетом (6) соотношения (7) приобретают вид

$$\sum_{T\ni i} \frac{p_T q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)} = q_i, \quad i \in M. \tag{8}$$

Сократив обе части на q_i, положительность которого следует из положительности p_i и того, что никакой символ a_j, $j \neq i$, не может на выходе нечеткого канала превратиться в a_i, приходим к уравнениям правдоподобия (2). По теореме 1 они имеют единственное решение, задающее распределение максимального правдоподобия.

Теорема доказана.

Чтобы говорить о информации в наблюдаемой нечеткой последовательности с эмпирическим распределением P о неизвестной исходной четкой последовтельности, необходимо сделать некоторые допущения о распределении Q на входе канала и совместном распределении PQ (или, что эквивалентно, — о условном распределении $Q|P$, т. е. условных вероятностях $p(i|T)$). Будем считать, что

- распределение вероятностей на входе канала является распределением $\hat{Q}$ максимального правдоподобия,
- условное распределение $\hat{Q}|P$ согласовано с каналом.

Эти распределения определены однозначно и при подстановке (6) в (5) дают значение условной энтропии

$$\mathcal{H}(\hat{Q}|P) = -\sum_T p_T \sum_{i\in T} \frac{\hat{q}_i p(T|i)}{\sum_{j\in T}\hat{q}_j p(T|j)} \log \frac{\hat{q}_i p(T|i)}{\sum_{j\in T}\hat{q}_j p(T|j)}. \tag{9}$$

Информация $\mathcal{I}(P;\hat{Q})$ выхода канала относительно входа в соответствии с обычным определением информации записывается в виде

$$\mathcal{I}(P;\hat{Q}) = -\sum_{i\in M} \hat{q}_i \log \hat{q}_i - \mathcal{H}(\hat{Q}|P). \tag{10}$$

Некоторые виды нечетких каналов

Более компактные выражения для условной энтропии $\mathcal{H}(\hat{Q}|P)$ и информации $\mathcal{I}(P;\hat{Q})$ могут быть получены при наложении ограничений на вид каналов. Нечеткий канал, в котором переходные вероятности $p(T|i)$, $i \in T$, представимы в виде $p(T|i) = \alpha_i d_T$, будем называть *каналом с разделенными искажениями*. Здесь α_i характеризует искажаемость четкого символа a_i, а d_T — возможность появления нечеткого символа a_T. Величины α_i и d_T определены с точностью до мультипликативной константы, ибо возможна одновременная замене всех α_i на $c\alpha_i$ и d_T на d_T/c. Для устранения неоднозначности будем считать $\min_i \alpha_i = 1$. Условия $\sum_T p(T|i) = 1$, $i \in M$, принимающие для данного канала вид $\alpha_i \sum_{T\ni i} d_T = 1$, выполнимы для произвольных $\alpha_i \geq 1$ и произвольных d_T, $|T| \neq 1$, столь малых, что при всех i имеет место $\alpha_i \sum_{T\ni i, T\neq\{i\}} d_T < 1$. Чтобы их удовлетворить, достаточно положить $d_{\{i\}} = 1/\alpha_i - \sum_{T\ni i, T\neq\{i\}} d_T$.

Для нечеткого канала с разделенными искажениями функция $\mathcal{H}(P,Q)$ из (1) преобразуется к виду

$$-\sum_T p_T \log \sum_{i\in T} \alpha_i q_i - \sum_T p_T \log d_T.$$

Вторая из участвующих здесь сумм не зависит от Q и распределение $\hat{Q}$ максимального правдоподобия, может быть найдено из условия минимизации первой суммы. С учетом этого будем для канала с разделенными искажениями в качестве $\mathcal{H}(P,Q)$ использовать эту сумму, т. е. считать

$$\mathcal{H}(P,Q) = -\sum_T p_T \log \sum_{i\in T} \alpha_i q_i. \tag{11}$$

Как и раньше, положим $\mathcal{H}(P) = \mathcal{H}(P,\hat{Q})$. Следующая теорема указывает величину средней информации в символе нечеткой последовательности, наблюдаемой на выходе канала с разделенными искажениями, о исходной четкой последовательности.

Теорема 3 *Для нечеткого канала с разделенными искажениями*

$$\mathcal{I}(P;\hat{Q}) = \mathcal{H}(P) + \sum_i \hat{q}_i \log \alpha_i.$$

Доказательство. Подставив в (9) значения $p(T|i) = \alpha_i d_T$ и сократив на d_T, находим

$$\mathcal{H}(\hat{Q}|P) = -\sum_T p_T \sum_{i\in T} \frac{\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j} \log \frac{\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j} = \sum\nolimits_1 - \sum\nolimits_2,$$

где

$$\begin{aligned}
\sum\nolimits_1 &= -\sum_T p_T \sum_{i\in T} \frac{\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j} \log(\alpha_i\hat{q}_i), \\
\sum\nolimits_2 &= -\sum_T p_T \sum_{i\in T} \frac{\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j} \log(\sum_{j\in T}\alpha_j\hat{q}_j).
\end{aligned}$$

Преобразуем первую сумму

$$\sum\nolimits_1 = -\sum_{T,i:\, i\in T} \frac{p_T\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j} \log(\alpha_i\hat{q}_i) = -\sum_i \hat{q}_i \log(\alpha_i\hat{q}_i) \sum_{T\ni i} \frac{p_T\alpha_i}{\sum_{j\in T}\alpha_j\hat{q}_j}.$$

Используя уравнения максимального правдоподобия (2), которые применительно к данному каналу приобретают вид

$$\sum_{T\ni i}\frac{p_T\alpha_i}{\sum_{j\in T}\alpha_j\hat{q}_j}=1,$$

получаем

$$\sum\nolimits_1=-\sum_i\hat{q}_i\log(\alpha_i\hat{q}_i)=-\sum_i\hat{q}_i\log\hat{q}_i-\sum_i\hat{q}_i\log\alpha_i.$$

Преобразования второй суммы дают

$$\sum\nolimits_2=-\sum_T p_T\frac{\sum_{i\in T}\alpha_i\hat{q}_i}{\sum_{j\in T}\alpha_j\hat{q}_j}\log(\sum_{j\in T}\alpha_j\hat{q}_j)=-\sum_T p_T\log(\sum_{j\in T}\alpha_j\hat{q}_j)=\mathcal{H}(P,\hat{Q}).$$

Из полученных выражений для $\sum_1$ и $\sum_2$, учитывая

$$\mathcal{H}(P,\hat{Q})=\mathcal{H}(P),$$

находим

$$\mathcal{H}(\hat{Q}|P)=-\sum_i\hat{q}_i\log\hat{q}_i-\sum_i\hat{q}_i\log\alpha_i-\mathcal{H}(P).$$

Откуда

$$\mathcal{I}(P;\hat{Q})=-\sum_i\hat{q}_i\log\hat{q}_i-\mathcal{H}(\hat{Q}|P)=\mathcal{H}(P)+\sum_i\hat{q}_i\log\alpha_i.$$

Теорема доказана.

Нечеткий канал, в котором переходные вероятности $p(T|i)$ не зависят от i, будем называть *равномерным*. Это частный случай канала с разделенными искажениями при $\alpha_i=1$ для всех $i\in M$. Заметим, что равномерный канал не является симметричным в смысле [3], поскольку структура множеств T, содержащих i, для разных i различна. Функция (11) для равномерного канала приобретает вид

$$\mathcal{H}(P,Q)=-\sum_T p_T\log\sum_{i\in T}q_i. \tag{12}$$

Пусть, как и раньше, $\mathcal{H}(P)=\mathcal{H}(P,\hat{Q})$. Отметим, что функция $\mathcal{H}(P)$ не использует характеристик канала, а определяется лишь набором параметров P рассматриваемой нечеткой последовательности. В работе [1] (см. также [2]) показано, что в задачах сжатия нечетких данных так определенная функция $\mathcal{H}(P)$ играет роль энтропии Шеннона. Эта функция встречается также в [4], где она введена для других целей. Из теоремы 1 вытекает следующее утверждение.

Следствие 1 *Для равномерного канала*

$$\mathcal{I}(P;\hat{Q})=\mathcal{H}(P).$$

Частным случаем равномерного канала является *канал со стиранием*. Он имеет четкий входной алфавит A_0, и нечеткий выходной алфавит $A=A_0\cup\{*\}$, где $*$ — неопределенный символ (в обозначениях данной работы он совпадает с a_M). Вероятность $p(*|i)$ перехода символа a_i в $*$ (стирания) одинакова для всех $i\in M$. Очевидно, стирающий канал симметричен в смысле [3].

Пусть параметрами наблюдаемой на выходе стирающего канала последовательности являются n_i, $i\in M$, и n_*. Обозначим через N число $\sum_{i\in M}n_i$ нестертых символов.

Следствие 2 *Для канала со стиранием*

$$\mathcal{I}(P|\hat{Q})=-\sum_{i\in M}\frac{n_i}{n}\log\frac{n_i}{N}.$$

Доказательство. Уравнения максимального правдоподобия применительно к каналу со стиранием приобретают вид

$$\frac{n_i}{nq_i}+\frac{n_*}{n\sum_{i\in M}q_i}=1.$$

Учитывая, что $\sum_{i\in M}q_i=1$, находим $q_i=n_i/(n-n_i)=n_i/N$. Подстановка этих значений в (12) дает

$$\mathcal{H}(P)=-\sum_{i\in M}\frac{n_i}{n}\log\frac{n_i}{N}-\frac{n_*}{n}\log\Big(\frac{1}{n}\sum_{i\in M}n_i\Big)=-\sum_{i\in M}\frac{n_i}{n}\log\frac{n_i}{N}.$$

Остается сослаться на следствие 1. Утверждение доказано.

Количество информации во всей наблюдаемой последовательности (о исходной четкой последовательности) может быть представлено в виде

$$n\mathcal{I}(P|\hat{Q}) = -\sum_{i\in M} n_i \log \frac{n_i}{N} = N\log N - \sum_{i\in M} n_i \log n_i.$$

Вычисление информационных характеристик нечетких последовательностей

В выражениях для условной энтропии $\mathcal{H}(\hat{Q}|P)$ (9) и информации $\mathcal{I}(P;\hat{Q})$ (10) нечетких последовательностей участвует распределение $\hat{Q}$ максимального правдоподобия. Опишем итеративную процедуру нахождения $\hat{Q}$ в общем случае. При этом под $\mathcal{H}(P,Q)$ будем понимать функцию общего вида (1).

Уравнения (2) максимального правдоподобия после домножения на q_i приобретают вид (8). На множестве $\mathcal{Q}^+ = \{Q = (q_0,\ldots,q_{m-1}) \mid q_0 > 0,\ldots,q_{m-1} > 0,\ q_0+\ldots+q_{m-1} = 1\}$ введем оператор $Q' = \mathcal{F}(Q)$, положив

$$q'_i = \sum_{T\ni i} \frac{p_T q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)}, \quad i \in M. \tag{13}$$

Из этого выражения видно, что при каждом $i \in M$ компонента q'_i содержит слагаемое p_i, поэтому $q'_i \geq p_i > 0$. Кроме того, подобно (2) устанавливается равенство $\sum_{i\in M} q'_i = 1$. Это означает, что $\mathcal{F}$: $\mathcal{Q}^+ \to \mathcal{Q}^+$. Более того, $\mathcal{F}(Q)$ содержится в замкнутом множестве

$$\mathcal{Q}^+_{p_0\ldots p_{m-1}} = \{Q \mid q_0 \geq p_0,\ldots,q_{m-1} \geq p_{m-1},\ q_0+\ldots+q_{m-1} = 1\}. \tag{14}$$

Лемма 1 *Распределение $\hat{Q}$ максимального правдоподобия является единственной неподвижной точкой оператора $Q' = \mathcal{F}(Q)$ на $\mathcal{Q}^+$.*

Доказательство. Неподвижные точки оператора $\mathcal{F}$ удовлетворяют равенствам (8), которые в силу положительности всех q_i эквивалентны уравнениям максимального правдоподобия (2). По теореме 1 их решение единственно.

Лемма доказана.

Лемма 2 *Для произвольного $Q \in \mathcal{Q}^+$ выполнено неравенство*

$$\mathcal{H}(\mathcal{F}(Q)) \leq \mathcal{H}(Q),$$

в котором равенство имеет место тогда и только тогда, когда Q — неподвижная точка оператора $\mathcal{F}$.

Доказательство. Положим $Q' = \mathcal{F}(Q)$ и рассмотрим разность

$$\Delta\mathcal{H} = \mathcal{H}(P,Q') - \mathcal{H}(P,Q) = -\sum_T p_T \log \sum_{i\in T} \frac{q'_i p(T|i)}{\sum_{j\in T} q_j p(T|j)}.$$

В силу (13) она преобразуется к виду

$$\Delta\mathcal{H} = -\sum_T p_T \log \sum_{i\in T}\sum_{S\ni i} \frac{p_S p(S|i)}{\sum_{l\in S} q_l p(S|l)} \cdot \frac{q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)}.$$

Учитывая, что

$$\sum_{i\in T} \frac{q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)} = 1,$$

и применяя неравенство Иенсена к выпуклой функции $(-\log x)$, получаем

$$\Delta\mathcal{H} \leq -\sum_T p_T \sum_{i\in T} \frac{q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)} \log \sum_{S\ni i} \frac{p_S p(S|i)}{\sum_{l\in S} q_l p(S|l)}.$$

Переставляя суммы, приходим к

$$\Delta\mathcal{H} \le -\sum_i \sum_{T\ni i} p_T \frac{q_i p(T|i)}{\sum_{j\in T} q_j p(T|j)} \log \sum_{S\ni i} \frac{p_S p(S|i)}{\sum_{l\in S} q_l p(S|l)}.$$

С учетом (13) и неравенства $-\sum_i q'_i \log q'_i \le -\sum_i q'_i \log q_i$ это дает

$$\Delta\mathcal{H} \le -\sum_i q'_i \log \frac{q'_i}{q_i} = -\sum_i q'_i \log q'_i + \sum_i q'_i \log q_i \le 0.$$

Откуда $\mathcal{H}(\mathcal{F}(Q)) \le \mathcal{H}(Q)$. Если имеет место равенство, то равенства возникают на всех предыдущих шагах и, в частности, $-\sum_i q'_i \log q'_i + \sum_i q'_i \log q_i = 0$. Но это возможно лишь при совпадении Q и $Q' = \mathcal{F}(Q)$, т. е. когда Q — неподвижная точка оператора $\mathcal{F}$.

Лемма доказана.

Теорема 4 *Для любого $Q^{(0)} \in \mathcal{Q}^+$ последовательность $Q^{(\nu)} = \mathcal{F}(Q^{(\nu-1)})$, $\nu = 1, 2, \ldots$, сходится к точке $\hat{Q}$ — распределению максимального правдоподобия.*

Доказательство. Рассмотрим произвольную предельную точку $\hat{Q}$ последовательности $Q^{(\nu)}$ и сходящуюся к ней подпоследовательность $Q^{(\nu_k)}$. Точка $\hat{Q}$ принадлежит замкнутому множеству (14) и, следовательно, — множеству $\mathcal{Q}^+$. В силу того, что $\mathcal{H}(P, Q^{(\nu_k)}) \to \mathcal{H}(P, \hat{Q})$ и величина $\mathcal{H}(P, Q^{(\nu)})$ не возрастает по ν (лемма 2), выполнено

$$\mathcal{H}(P, Q^{(\nu)}) \to \mathcal{H}(P, \hat{Q}). \tag{15}$$

Поскольку $Q^{(\nu_k+1)}$ сходится к точке $\mathcal{F}(\hat{Q})$, величина $\mathcal{H}(P, Q^{(\nu_k+1)})$ стремится к $\mathcal{H}(P, \mathcal{F}(\hat{Q}))$. С учетом (15) это дает $\mathcal{H}(P, \mathcal{F}(\hat{Q})) = \mathcal{H}(P, \hat{Q})$. Отсюда на основе леммы 2 заключаем, что $\hat{Q}$ является неподвижной точкой оператора $\mathcal{F}$. По лемме 1 неподвижная точка единственна и, следовательно, последовательность $Q^{(\nu)}$ имеет единственную предельную точку $\hat{Q}$, к которой она и сходится. Согласно лемме 1 эта точка соответствует распределению максимального правдоподобия.

Теорема доказана.

После нахождения $\hat{Q}$ могут быть в соответствии с (9) и (10) вычислены величины $\mathcal{H}(\hat{Q}|P)$ и $\mathcal{I}(P; \hat{Q})$.

Список литературы

1. Шоломов Л. А. Сжатие частично определенной информации // Нелинейная динамика и управление. Вып. 4. - М.: Физматлит, 2004. С. 385–399.

2. Шоломов Л. А. Кодирование частично-определенных дискретных источников без памяти // Доклады Академии наук. 2004. Т. 397, N 2. С. 178–180.

3. Галлагер Р. Теория информации и надежная связь. - М.: Советское радио, 1974.

4. Питерсон У. Коды, исправляющие ошибки - М.: Мир, 1964.

5. Бонгард М. М. О понятии "полезная информация" // Проблемы кибернетики. Вып. 9. - М.: Физматгиз, 1963. С. 71–102.

Поиск частых подпоследовательностей

Шуткин Юрий,

Механико-математический факультет МГУ им. Ломоносова

E-mail: yuriish@yandex.ru

В настоящее время все чаще возникает необходимость анализа огромных массивов данных (Data Mining). В данной статье рассматривается одна из задач Data Mining — поиск последовательности фиксированной длины, имеющей наибольшее число вхождений в исходную последовательность. Приводятся некоторые алгоритмы для ее решения. Дается оценка сложности этих алгоритмов.

Постановка задачи

Дана последовательность S длины n, состоящая из 0 и 1. Пусть эта последовательность строилась по закону: ноль появляется с вероятностью p_0, а единица — с вероятностью p_1.

Нужно определить, какая подпоследовательность (слово) длины l является наиболее часто встречающейся. Обозначим за N_β число вхождений слова β в последовательность S. Надо найти (хотя бы одну) подпоследовательность β^* такую, что

$$N_{\beta^*} = \max_{\beta \in \{0,1\}^l} N_\beta.$$

В [1,2] рассматривается задача нахождения всех слов, частота которых не меньше заданного порога. Мы же ограничимся нахождением только самой частой.

Будем говорить, что слово β длины l находится на месте i в исходной последовательности S, если $S(i+j-1) = \beta(j),\ j = 1, \ldots, l$.

Ставится задача минимизировать сложность — количество обращений к исходной последовательности. Обращением будем считать сравнение какого-то отрезка длины l исходной последовательности с другой последовательностью из нулей и единиц длины l.

Просмотром исходной последовательности будем называть последовательное обращение ко всем отрезкам длины l. Далее будем оценивать именно количество просмотров.

Для того, чтобы посчитать число вхождений произвольной подпоследовательности, нам понадобится n обращений, или один просмотр исходной последовательности.

Разрастающийся алгоритм (РА)

Понятно, что простой перебор не даст хорошего результата.

Приведем алгоритм, который по сравнению с перебором позволит сократить количество обращений к исходной последовательности. За основу взят алгоритм GSP (Generated Sequence Pattern), вычисляющий частоты слов последовательно, начиная с самых коротких. Описание алгоритма можно найти в [2].

Разрастающийся алгоритм основан на простых равенствах:

$$N_{0\beta} + N_{1\beta} = N_\beta$$

$$N_{\beta 0} + N_{\beta 1} = N_\beta,$$

где слово 0β есть просто слово β с дописанным в начало нулем.

Сначала находим число вхождений слов длины один: N_0 и N_1. Причем т.к. $N_0 + N_1 = |S|$, непосредственно считать число вхождений придется только для одной из них, а второе (пусть N_1) можно вычислить как $|S| - N_0$.

Пусть все частоты последовательностей длины i у нас известны и хранятся в памяти. Посчитаем то же самое для последовательностей длины $i+1$. Выпишем четыре равенства

$$N_{0\beta 0} + N_{1\beta 0} = N_{\beta 0}$$

$$N_{0\beta 0} + N_{0\beta 1} = N_{0\beta}$$

$$N_{0\beta 1} + N_{1\beta 1} = N_{\beta 1}$$

$$N_{1\beta 0} + N_{1\beta 1} = N_{1\beta}$$

для каждого β длины $i-1$ (β может быть пустым). Таким образом, для того, чтобы найти 4 числа $N_{0\beta 0},\ N_{0\beta 1},\ N_{1\beta 0},\ N_{1\beta 1}$ достаточно посчитать число вхождений для одного из этих слов, а остальные вычислить, используя $N_{0\beta},\ N_{1\beta},\ N_{\beta 0},\ N_{\beta 1}$. Итого, если мы захотим узнать все N_β для β таких, что $|\beta| = l$, то нам понадобится $1 + 1 + 2 + \cdots + 2^{l-2} = 2^{l-1}$ просмотров исходной последовательности.

Вероятностный подход

Есть другой поход к решению поставленной задачи.

Если нам достаточно получить решение с какой-то вероятностью, а не обязательно точное, то можно сразу, зная закон распределения, оценить вероятность того, что выбранное нами слово будет самым частым, или выделить какое-то множество потенциально частых последовательностей.

Для события $A_{\beta,i}$, при котором на i-м месте в исходной последовательности стоит слово β имеет место формула

$$P(A_{\beta,i}) = p_0^{\beta_0} p_1^{\beta_1},\ \forall i.$$

Пусть исходная последовательность порождается по закону 0 с вероятностью p_0, 1 с вероятностью p_1. Оценим такую вероятность

$$P(N_{\beta_1} \geq N_{\beta_2}),$$

где N_{β_1} и N_{β_2} - соответственно частоты слов β_1 и β_2 в исходном слове.

$$P(A) = \sum_S I_A(S) \cdot P(S).$$

$$I_A(S) = \begin{cases} 1, & \text{если событие } A \text{ выполняется в последовательности } S \\ 0, & \text{иначе.} \end{cases}$$

Для того, чтобы посчитать вероятность того, что одно слово встречается чаще другого, представим, что частота каждого слова — это случайная величина, и посчитаем ее матожидание и дисперсию. Потом, используя неравенство Чебышева получим интересующую нас оценку.

Пользуясь полученной формулой для вероятности появления слова на i-м месте получаем

$$E(N_\beta) = \sum_{i=1}^{n} p_0^{\beta_0} p_1^{\beta_1} = n p_0^{\beta_0} p_1^{\beta_1} = n p_\beta. \tag{16}$$

Мы обозначили $p_0^{\beta_0} p_1^{\beta_1}$ через p_β для удобства.

Дисперсию вычисляем по определению

$$D_\beta = n(n - 2l + 1)p_\beta^2 + 2np_\beta \sum_{\lambda=1}^{l-1} p_\beta^{(\lambda)} + E_\beta - E_\beta^2,$$

где $p_\beta^{(\lambda)}$ — некоторая характеристика слова β.

$$D_\beta = E_\beta \cdot \left(1 + (1 - 2l)p_\beta + 2\sum_{\lambda=1}^{l-1} p_\beta^{(\lambda)}\right) = E_\beta \cdot C_\beta. \tag{17}$$

Теперь мы знаем матожидание и дисперсию нашей случайной величины N_β. С помощью них оценим величину $P(N_{\beta_1} \geq rN_{\beta_2})$. Пусть $E_{\beta_1} = sE_{\beta_2}$. Тогда, делая простые преобразования и пользуясь неравенством Чебышева, получаем

$$P(N_{\beta_1} \geq rN_{\beta_2}) \geq \left(1 - \frac{4r^2 C_{\beta_2}}{(s-r)^2 E_{\beta_2}}\right)\left(1 - \frac{4s^2 C_{\beta_1}}{(s-r)^2 E_{\beta_1}}\right). \tag{18}$$

Можем заметить, что при $n \to \infty$ эта вероятность стремится к единице, если $s > r$, что и должно было получиться. Причем стремится она как $1/n$ к нулю (по порядку).

Используя предыдущую оценку, можно не прибегая ни к каким алгоритмам сразу оценить вероятность того, что некоторое слово встретиться чаще всех остальных, или хотя бы выделить небольшое множество слов, в котором потом искать, например, перебором.

Например, если вероятность появления единицы равна p_1, а вероятность нуля — p_0, то вероятность того, что самое частое слово встретится среди тех, у которых меньше z нулей (или, что то же самое, больше $l - z$ единиц) может быть оценена снизу

$$P(N_\beta \geq N_{\gamma_i}) \geq \left(1 - \frac{4C_\beta}{np_\beta(1 - \kappa^{-z})^2}\right) \prod_{i=1}^{t} \left(1 - \frac{4C_{\gamma_i}}{np_{\gamma_i}(\kappa^{\gamma_{i0}} - 1)^2}\right), \tag{19}$$

где $\kappa = \frac{p_1}{p_0}$, а $\gamma_i,\ i = 1, \ldots, t$ - слова, в которых больше z нулей.

Таким образом, если мы хотим с заданной вероятностью (например $1 - \varepsilon$) найти самое частое слово, то для этого достаточно подобрать такое число z, что

$$\left(1 - \frac{4C_\beta}{np_\beta(1-\kappa^{-z})^2}\right)\prod_{i=1}^{t}\left(1 - \frac{4C_{\gamma_i}}{np_{\gamma_i}(\kappa^{\gamma_{i0}}-1)^2}\right) \geq (1-\varepsilon).$$

Это будет означать, что вероятность того, что самое частой слово среди тех, у которых много нулей, не больше ε, то есть останется перебрать те слова, у которых нулей меньше z.

Заметим, что t не зависит от n ($t = \sum\limits_{k=0}^{l-z} C_l^k$), поэтому при стремлении $n \to \infty$, вероятность в (19) стремится к 1.

Итак, мы получили оценку вероятности того, что самое частое слово встретится среди тех, у которых меньше z нулей. И эта вероятность стремится к единице при $n \to \infty$.

Автор выражает благодарность Гасанову Э.Э. за постановку задачи и помощь в получении результатов.

Список литературы

1. Tumasonis R., Dzemyda G.: A probabilistic algorithm for mining frequent sequences. In Proceedings of Eighth East-European Conference on Advances in Databases and Information Systems - ADBIS'04, Budapest, Hungary (2004) 89-98

2. Han J., Pei J., Yin Y: Mining frequent patterns without candidate generation. Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), Dallas TX (2000) 1-12

Реализация булевых функций с помощью информационных графов

Шуткин Юрий,
Механико-математический факультет МГУ им. Ломоносова
E-mail: yuriish@yandex.ru

Рассматривается задача реализации булевых функции с помощью информационных графов. Получено точное значение функции Шеннона сложности в классе древовидных информационных графов. Получен порядок сложности реализации информационными графами для почти всех булевых функций.

Постановка задачи

Ориентированным информационным графом G будем называть сеть с одним входом и одним выходом, в которой все контакты ориентированы от входа к выходу (по сути определение совпадает с определением ориентированной контактной схемы в [1], с той лишь разницей, что сложность графа задается по-другому). Вход будем называть *корневой вершиной* графа, а выход — *конечной вершиной*. Контакты будем называть *ребрами* графа, а приписанные им переменные — *предикатами*. Считается, что контакт вида x_i^σ проводит запрос α тогда и только тогда, когда $\alpha_i = \sigma$.

Говорим, что запрос α проходит из вершины v_1 в вершину v_2, если существует ориентированный путь из v_1 в v_2, такой, что все ребра этого пути проводят запрос α (обозначаем $v_2 \in \theta_{v_1}(\alpha)$).

Информационный граф G реализует булеву функцию f если любой набор α, на котором функция принимает значение 1, проходит из начальной вершины v_0 графа G в конечную вершину w, и любой набор β, на котором функция принимает значение 0, не проходит. Множество графов, реализующих функцию f обозначим через $U(f)$.

Более общее определение информационного графа и его функционирования можно найти в [2].

Количество предикатов, вычисленных на запросе α в графе G считается следующим образом. Помечаются все вершины, в которые проходит запрос α. Считаем, что в вершине v вычисляются те предикаты, которые приписаны ребрам, выходящим из этой вершины. Общее количество вычисленных предикатов на запросе α — сумма по всем помеченным вершинам вычисленных в них предикатов.

$$L(G,\alpha) = \sum_{v \in \theta_{v_0}(\alpha)} \psi(v),$$

где $\psi(v)$ — количество ребер, выходящих из v (степень исхода вершины v).

Пусть на множестве запросов введено вероятностное пространство. Сложностью информационного графа назовем величину

$$L(G) = \sum_{\alpha \in \{0,1\}^n} L(G,\alpha)P(\alpha) = E_\alpha\big(L(G,\alpha)\big),$$

где $P(\alpha)$ — вероятность запроса α в нашем вероятностном пространстве.

Сложностью функции назовем минимальную сложность графа, реализующего эту функцию.

$$L(f) = \min_{G \in U(f)} L(G).$$

Функцией Шеннона сложности реализации булевых функций n переменных информационными графами назовем максимальную сложность функций из $P_2(n)$.

$$L^{Sh}(n) = \max_{f \in P_2} L(f).$$

Определим также сложность реализации булевой функции с помощью деревьев и функцию Шеннона для древовидных графов.

Так как конечная вершина у графа должна быть одна, а у деревьев их много, то вместо деревьев будем использовать понятие квазидерева.

Квазидерево — это граф, полученный из соответствующего дерева путем отождествления листьев и объявления полученной вершины конечной вершиной графа.

$$L_D(f) = \min_{G \in D(f)} L(G),$$

где $D(f)$ — множество квазидеревьев, реализующих функцию f.

$$L_D^{Sh}(n) = \max_{f \in P_2} L_D(f).$$

Верхняя оценка

Построим граф, реализующий произвольную булеву функцию, следующим образом (приведенный метод схож с методом каскадов для контактных схем, см. [1]).

Основная идея — разложить функцию сначала по первой переменной, потом по второй, и так далее.

$$f(x_1, x_2, \ldots, x_n) = x_1 f(1, x_2, \ldots, x_n) \vee \bar{x}_1 f(0, x_2, \ldots, x_n) = \ldots$$

Граф, реализующий функцию f будет таким: из корня выходит два ребра — x_1 и $\bar{x}_1$. Дальше на концах этих двух ребер строим граф, реализующий соответственно функции $f(1, x_2, \ldots, x_n)$ и $f(0, x_2, \ldots, x_n)$. И так далее, пока не дойдем до функции одной переменной. А ее реализуем не более чем одним ребром.

Легко убедиться по индукции, что сложность такого графа не больше $2n - 1$.

Таким образом, доказана

Теорема 1 *Сложность реализации булевой функции n переменных с помощью информационного графа не превышает $2n - 1$.*

Нижняя оценка функции Шеннона

Рассмотрим реализацию булевых функций информационными деревьями.

С помощью квазидеревьев реализовать функцию $f = \bigoplus_{i=1}^{n} x_i$ со сложностью, меньшей $2n-1$, не удастся.

Эта функция обладает тем свойством, что сколько бы переменных мы не фиксировали (не все, конечно), полученная подфункция не будет константой. Из этого будет следовать, что пути из начальной вершины в конечную будут длины не меньше n. Несложными преобразованиями можно добиться длины всех путей не больше n.

Таким образом, получаем, что все пути оптимального дерева, реализующего функцию $\bigoplus_{i=1}^{n} x_i$, имеют длину n.

Опять же из свойств функции следует, что на каждом i-м ярусе дерева вершин будет не меньше 2^i (не считая последнего).

Вероятность прохождения запроса в любую вершину i-го яруса равна 2^{-i}.

Суммируя по всем вершинам получаем, что сложность функции f равна $2n-1$.

Вспоминая верхнюю оценку, получаем

Теорема 2 *Для функции Шеннона в классе деревьев справедливо равенство*

$$L_D^{Sh}(n) = 2n - 1.$$

В классе информационных графов нижняя оценка немного слабее, тем самым, оценен только порядок функции Шеннона.

Теорема 3 *Для функции Шеннона в классе информационных графов справедливо неравенство*

$$\frac{3n-1}{2} \le L^{Sh}(n) \le 2n - 1.$$

Порядок сложности для почти всех функций

Выделим некоторый класс функций, для которых будет получена нижняя оценка сложности и покажем, что на самом деле это почти все функции.

Введем на множестве всех функций две функции $\zeta_0(f)$ и $\zeta_1(f)$.

$\zeta_j(f) = k$ тогда и только тогда, когда существует набор $i_1, \dots, i_k$ и набор $\alpha_{i_1}, \dots, \alpha_{i_k}$ такой, что $f(*, \dots, *, \alpha_{i_1}, *, \dots, *, \alpha_{i_k}, *, \dots, *) = j$, т.е. полученная подфункция является тождественной константой j. Причем для $l = k-1$ такого набора $\alpha_{i_1}, \dots, \alpha_{i_l}$ уже не существует.

Нетрудно показать, что сложность любой функции f удовлетворяет неравенству

$$L(f) \ge 2^{-n}\Big(\zeta_0(f)N_0(f) + \zeta_1(f)N_1(f)\Big),$$

где $N_{[0,1]}(f)$ — количество нулей и единиц функции f.

Обозначим через $\zeta(f)$ минимальную из $\zeta_0(f)$ и $\zeta_1(f)$, и назовем полученную функцию *степенью существенности* функции f.

Тогда предыдущее неравенство можно переписать в немного ослабленном виде

$$L(f) \ge \zeta(f).$$

Количество функций, для которых $\zeta(f) \le s$, не больше $2^{2^n - 2^{n-s}+s+1} \cdot C_n^s$.

Фиксируем $s = (1-\varepsilon)n$.

Легко видеть, что при $n \to \infty$ количество функций таких, что $\zeta(f) \le (1-\varepsilon)n$, есть $o(2^{2^n})$, и имеет место

Теорема 4 *Для почти всех булевых функций f от n переменных выполнено*

$$L(f) \asymp n.$$

Автор выражает благодарность Гасанову Э.Э. за постановку задачи и помощь в исследовании.

Список литературы

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

2. Гасанов Э.Э., Кудрявцев В.Б. Теория хранения и поиска информации: Изд-во Физматлит, 2002

Оглавление

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ"(23-27 октября 2006 г.), том 1, часть 2. Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В., 2006 г., 168 с.

Подписано в печать 29.09.2006.
Формат 60×90 1/16. Объем 11.5 п.л.
Заказ 18. Тираж 200 экз.

Издательство ЦПИ при механико–математическом факультете МГУ
г. Москва, Воробьевы горы.
Лицензия на издательскую деятельность ИД № 04059 от 20.02.2001 г.

Отпечатано на типографском оборудовании механико-математического факультета