Репрезентативность случайно сгенерированных недетерминированных конечных автоматов с точки зрения соответствующих базисных автоматов[1]

Б. Ф. Мельников, д. ф.-м. н.,
С. В. Пивнева, к. п. н.,
О. А. Рогова,
Тольяттинский государственный университет
B.Melnikov@tltsu.ru, tlt.swetlana@rambler.ru, rogovaolgatlt@yandex.ru

В статье рассматриваются варианты случайной генерации недетерминированных конечных автоматов. Эти варианты являются авторскими усовершенствованиями известного алгоритма Ван Зиджл, основанного на случайных битовых потоках. Также рассмотрен авторский подход к проверке репрезентативности входных данных на основе приведенных алгоритмов генерации автоматов. Один из рассматриваемых в статье критериев проверки репрезентативности — специально выбираемые характеристики эквивалентного базисного автомата.

Результаты проведенных вычислительных экспериментов показывают, что предлагаемые в статье алгоритмы случайной генерации дают недетерминированные конечные автоматы, более подходящие для решения задач выбранной предметной области.

Ключевые слова: недетерминированный конечный автомат, базисный автомат, алгоритмы случайной генерации, репрезентативность.

1. Введение

Понятие “репрезентативность” в статистике — это основное требование к выборочной совокупности, заключающееся в соответствии ее характеристик соответствующим характеристикам генеральной совокупности, из которой с соблюдением определенных правил и отобрана выборочная. Однако, если воспользоваться более широким, но редко употребляемым понятием репрезентативности, как “возможность воспроизвести представление о целом по его части”, то оно становиться сложным, т. к. оно описывает процесс, а не объект (выборку). Репрезентативность – величина измеряемая, которая может быть определена “ошибкой репрезентативности”, т. е. разностью между специально выбираемыми характеристиками выборочной и генеральной совокупностей. Однако фактическая (действительная) величина указанной разности остается неизвестной —

вследствие чего мерой репрезентативности обычно служит определяемая по правилам математической статистики ее вероятная величина (или среднее квадратичное ее возможных значений). Случайная генерация комбинаторных структур позволяет проверять алгоритмы, основанные на этой структуре, и исследовать поведение этих структур. А проверка репрезентативности сгенерированных структур проводится нами с помощью специально подобранных некоторых статистических критериев. Сгенерированные объекты адекватны тем потребностям, которые возникают в реальных задачах. Например, при описании контекстно-свободных языков с помощью конечных автоматов. В реальных задачах требуется достаточно большое количество состояний конечного автомата, поэтому необходимо генерировать недетерминированные конечные автоматы с целью применения к ним конкретных характеристик, которые впоследствии применяются, например, в некоторых алгоритмах LR-анализа. Конечные автоматы используются также в лексических анализаторах, в лексическом анализе при компиляции языков программирования и при трансляциях, которые обеспечивают человеко-машинный интерфейс, и тестировании программного обеспечения на основе моделей.

2. Недетерминированные конечные автоматы — предварительные сведения

Будем использовать следующие обозначения. Пусть

$$K = (Q, \Sigma, \delta, S, F) \tag{1}$$

— некоторый конечный автомат (недетерминированный автомат Рабина-Скотта), определяющий регулярный язык $L = \mathcal{L}(K)$. Здесь Q – множество состояний, а $S \subset Q$ и $F \subset Q$ — множества стартовых и финальных состояний соответственно. Мы будем рассматривать автомат без ϵ-переходов, т. е. будем рассматривать функцию переходов δ автомата (1) в виде $\delta : Q \times \Sigma \to \mathcal{P}(Q)$; записью $\mathcal{P}(Q)$ обозначено множество всех подмножеств множества Q.

Через $\mathcal{L}_K^{in}(q)$ и $\mathcal{L}_K^{out}(q)$ обозначим входной и выходной языки состояния q — т. е. языки, определяемые автоматами $(Q, \Sigma, \delta, S, \{q\})$ и $(Q, \Sigma, \delta, \{q\}, F)$ соответственно.

Для автомата (1) и пары $p, q \in Q$, будем также использовать следующие обозначения. Если $\delta(p, a) \in r$, то будем писать $p \xrightarrow[\delta]{a} r$

или $p \xrightarrow[K]{a} r$. Будем иногда опускать нижние индексы δ и K – если автомат подразумевается. Аналогичные обозначения будем использовать и для слов. Например, мы будем писать $p \xrightarrow[K]{u} r$, если существует путь из p в r, помеченный словом $u \in \Sigma^*$; при этом p будет опускаться, когда мы рассматриваем путь из некоторого входа, и т. п.

Зеркальный автомат для заданного автомата (1) будем обозначать K^R — т. е. $K^R = (Q, \Sigma, \delta^R, F, S)$, где $q' \xrightarrow[\delta^R]{a} q''$ если и только если $q'' \xrightarrow[\delta]{a} q'$. Очевидно, что K^R определяет язык L^R.

Для рассматриваемого языка L будем обозначать эквивалентный ему канонической автомат записью $\widetilde{L}$. Далее будем считать, что автоматы $\widetilde{L}$ и $\widetilde{L^R}$ для рассматриваемого языка L следующие:

$$\widetilde{L} = (Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi) \text{ и } \widetilde{L^R} = (Q_p, \Sigma, \delta_p, \{s_p\}, F_p).$$

Рассмотрим некоторые определения и факты из [1–4], используемые далее в статье. Определим две специальные функции ϕ^{in} и ϕ^{out} — т. н. функции разметки состояний автомата (1), помечающие каждое состояние некоторыми подмножествами состояний $\widetilde{L}$ и $\widetilde{L^R}$. Полагаем $\phi^{in}_K(q) \ni A$ тогда и только тогда, когда существует слово $u \in \Sigma^*$, такое что $\xrightarrow[K]{u} q$ и $\xrightarrow[BA(L)]{u} A$; аналогично для функции ϕ^{out}_K. Заметим, что эти определения конструктивны, т. е. задают *алгоритмы* построения данных функций ϕ^{in} и ϕ^{out}.

Далее рассмотрим специальное бинарное отношение $\#$ на множестве $Q_\pi \times Q_\rho$; отметим, что оно относится к заданному языку, а не к конкретным автоматам для его определения. Проще всего указанное отношение может быть получено при построении функции ϕ^{in}, рассматриваемой для автомата, зеркального к каноническому для заданного языка. Отметим, что пары состояний отношения $\#$ могут рассматриваться как состояния т. н. базисного автомата для данного языка (см. [2, 3] и др.). Кроме того, отношение $\#$ также формирует множество т. н. псевдогридов (псевдоблоков); мы можем считать, что каждый из них — некоторая пара (P, R), где $P \subset Q_\pi$, $R \subset Q_\rho$ и для каждой пары состояний $p \in P$ и $r \in R$ выполнено условие $p\#r$. Если для некоторого псевдоблока (P, R) мы не можем расширить ни P, ни R для получения нового псевдоблока, то будем называть этот псевдоблок гридом (блоком).

Каждый из этих псевдоблоков соответствует состоянию любого конкретного автомата для заданного языка. Более того, необходимым условием для определения данного языка конечным автоматом является то, что подмножество псевдоблоков, соответствующих множеству состояний рассматриваемого автомата, покрывает все элементы отношения $\#$[2]. В [2, 4] был приведен алгоритм, описывающий все возможные дуги автомата[3], состояния которого соответствуют данному множеству псевдоблоков.

3. Метод случайной генерации недетерминированных конечных автоматов

В нашей работе рассматривается случайный метод генерации недетерминированного конечного автомата Ван Зиджл, основанный на случайных битовых потоках [5]. С помощью этого метода с равновероятными битовыми потоками можно успешно сравнивать различные представления регулярных языков. Но структуры, сгенерированные этим методом, не соответствуют требованиям выбранной предметной области. Например, в автоматах, сгенерированных методом Ван Зиджл, сравнительно мало циклов, не устраивает также получаемая разреженность автомата. Поэтому был разработан другой алгоритм генерации недетерминированного конечного автомата, результатом работы которого являются автоматы, более соответствующие рассматриваемым характеристикам и подходящие для решения задач выбранной предметной области.

Опишем новый метод, используемый для случайной генерации недетерминированного автомата в выбранной предметной области. Отметим, что вариант этого метода был описан ранее в [6].

• Задан алфавит $\Sigma = \{1, \ldots, m\}$ и набор состояний $Q = 1, \ldots, n$.

• Произведены равновероятные битовые потоки размера mn^2; они описывают функцию перехода δ; возникновение бита отличного от нуля в положении $2 \times ((l-1) \times n^2 + (i-1) + j)$ обозначает существование перехода от состояния i в состояние j, помеченного l.

[2]Мы не будем рассматривать различные возможные версии *достаточных* условий этого факта. Конечно, такие условия могут быть просто получены построением эквивалентного канонического автомата — однако такое решение дает слишком неэффективный алгоритм.

[3]Также там описаны возможные стартовые и финальные состояния. Однако на самом деле в указанной статье рассматривались не автоматы, а специальные "автоматоподобные" структуры.

• Строится трехмерная матрица переходов. В ячейках матрицы записываются символы, которые можно прочитать при переходе из состояния q_i в состояние q_j.

• Есть единственное начальное состояние (вершина 1).

• Набор финальных состояний выбран случайно, каждое состояние имеет равный шанс на то, чтобы быть финальным.

4. Проверка репрезентативности сгенерированных структур

Для проверки репрезентативности сгенерированных структур нами применялись некоторые статистические критерии из приведенных в [7]. Если критерии $T_1, T_2, \ldots, T_n$ подтверждают, что последовательность ведет себя случайным образом, это не означает, что проверка с помощью T_{n+1}-го критерия будет успешной. Обычно к последовательности применяется около шести статистических критериев, и если она удовлетворяет *всем* этим критериям, то последовательность считается случайной.

Критерий "хи-квадрат". Проводим n независимых наблюдений, каждое наблюдение может принадлежать к одной из k категорий. Пусть p_s — вероятность того, что каждое наблюдение относится к категории s, пусть Y_s — число наблюдений, которые действительно относятся к категории s. Образуем статистику

$$V = \frac{1}{n} \sum_{s=1}^{k} \left(\frac{Y_s^2}{p_s} \right) - n.$$

Приемлемое значение статистики V можно определить по таблице, которая дает значения "χ^2-распределения с ν степенями свободы" для различных значений ν. Используется строка таблицы с $\nu = k - 1$, так как число "степеней свободы" равно $k - 1$, что на единицу меньше, чем число категорий. Если получаемое значение V меньше 1%-й точки или больше 99%-й точки, то эти числа отбрасываются как недостаточно случайные. Если V лежит между 1%- и 5%-й точками или между 95%- и 99%-й точками, то эти числа "подозрительны"; если же V лежит между 5%- и 10%-й точками или 90%- и 95%-й точками, числа можно считать "почти подозрительными". Проверка по χ^2-критерию проводится три раза и более с разными данными. Если по крайней мере два из трех результа-

тов оказываются подозрительными, то числа рассматриваются как недостаточно случайные.

Критерий равномерности. Проверяем равномерность распределения чисел. Выбирается число d. Для каждого r, $0 \le r < d$, подсчитывается число случаев, когда $Y_j = r$ для $0 \le j < n$, а затем применяется χ^2-критерий, принимая $k = d$ и вероятности $p_s = 1/d$ для каждой категории.

Критерий серий. Проверяем требование к последовательности, состоящее в том, чтобы пары последовательных чисел были равномерно распределены независимым образом. Подсчитываем число случаев, когда пара $(Y_{2j}, Y_{2j+1}) = (q, r)$ для $0 \le j < n$. Такая операция осуществляется для каждой пары целых чисел (q, r), таких, что $0 \le q, r < d$. Затем применяется χ^2-критерий к этим категориям, где $1/d^2$ — вероятность отнесения пары чисел к каждой из категорий.

Критерий интервалов. Этот критерий используется для проверки длины "интервалов" между появлением U_j на определенном отрезке. Если α и β — два действительных числа, таких что $0 \le \alpha < \beta \le 1$, то рассматриваем длины подпоследовательностей U_j, U_{j+1}, $, \ldots, U_{j+r}$, в которых U_{j+r} лежит между α и β, а другие U_s не лежат между этими числами. χ^2-критерий применяется при $k = t+1$ к значениям $count[0], count[1], \ldots, count[t]$ ($count[r]$ – число интервалов длины r) с использованием следующих вероятностей: $p_r = p(1-p)^r$ для $0 \le r \le t-1$; $\quad p_t = (1-p)^t$, где $p = \beta - \alpha$ — вероятность того, что $\alpha \le U_j < \beta$. Значения n и t выбираются так, чтобы ожидаемое значение $count[r]$ равнялось пяти или больше.

Покер-критерий (критерий разбиений). Рассматриваем n групп по пять последовательных целых чисел $\{Y_{5j}, Y_{5j+1}, \ldots, Y_{5j+4}\}$ для $0 \le j < n$ и проверяем, какие из следующих пяти категорий соответствуют пятеркам чисел. В общем случае можно рассматривать n групп k последовательных чисел и подсчитывать число групп из k чисел с r различными числами. Затем применяется χ^2-критерий, в котором используются вероятности того, что в группе r различных чисел

$$p_r = \frac{d(d-1)\ldots(d-r+1)}{d^k} \left\{ {k \atop r} \right\}, \qquad \left\{ {r \atop k} \right\} = \prod_{j=1}^{k} \frac{r+1-j}{j}.$$

Критерий собирания купонов. Рассматриваем последовательность $Y_0, Y_1, \ldots$ и находим длины отрезков $Y_{j+1}, Y_{j+2}, \ldots, Y_{j+r}$, со-

держащие "полный набор" целых чисел от 0 до $d-1$. Если дана последовательность целых чисел $Y_0, Y_1, \ldots$, таких что $0 \le Y_j < d$, то подсчитываются длины n последовательных "собравших купоны" отрезков ($count[r]$ — это число отрезков длины r для $d \le r < t$, а $count[t]$ — это число отрезков длиной $\ge t$). После того, как вычислены n длин, нужно применить χ^2-критерий к $count[d], count[d+1], \ldots, count[t]$ с $k = t - d + 1$. Соответствующие вероятности равны

$$p_r = \frac{d!}{d^r} \left\{ {r-1 \atop d-1} \right\},\ d \le r < t;\ p_t = 1- = \frac{d!}{d^{t-1}} \left\{ {t-1 \atop d} \right\}.$$

Критерий сериальной корреляции. Если задано n величин $U_0, U_1, \ldots, U_{n-1}$ и n других величин $V_0, V_1, \ldots, V_{n-1}$, то коэффициент корреляции между ними определяется следующим образом:

$$C = \frac{n\sum(U_jV_j) - (\sum U_j)(\sum V_j)}{\sqrt{(n\sum U_j^2 - (\sum U_j)^2)(n\sum V_j^2 - (\sum V_j)^2)}},\ 0 \le n.$$

Если коэффициент корреляции равен нулю или очень мал, то величины U_j и V_j будут независимы; если же значение коэффициента корреляции равно $+1$ или -1, это означает полную линейную зависимость.

5. Заключение

Описанные критерии были применены к сгенерированной последовательности чисел. Практическая реализация данных критериев показала, что сгенерированная последовательность чисел достаточно случайна, требования всех критериев были выполнены. Ранее, в предыдущей публикации авторов [6], было описано применение к сгенерированным недетерминированным конечным автоматам следующих характеристик:

1) разреженность автомата;

2) вложенность циклов;

3) минимальная длина пути от стартовой вершины до финальной, деленная на количество вершин.

В настоящее время авторы добавили две новые характеристики:

4) число вершин получаемого эквивалентного базисного автомата;

5) число дуг эквивалентного базисного автомата.

Программная реализация разработанного метода случайной генерации недетерминированного конечного автомата и проведенные вычислительные эксперименты показали, что сгенерированные структуры удовлетворяют требуемым характеристикам. А именно (по сравнению со стандартным вариантом метода Ван Зиджл):
1) описанный выше метод генерирует автоматы с уменьшенным средним числом дуг из вершины:

Количество состояний автомата	2	4	6	8	10	12	14	16	18	20
Метод Ван Зиджл	1	2	3	3,8	5	5,8	6,8	7,5	8,8	10,2
Разработанный метод	0	1,1	1,8	2,5	3,2	3,5	4	4,4	4,8	5,1

2) уровень вложенности циклов увеличен:

Количество состояний автомата	2	4	6	8	10	12	14	16	18	20
Метод Ван Зиджл	0	2	3,5	5	8	9	10	12	13	14
Разработанный метод	0	1	1	2	3	3	5	5	6	6

3) минимальная длина пути от стартовой вершины до финальной, деленная на количество вершин, уменьшилась:

Количество состояний автомата	2	4	6	8	10	12	14	16	18	20
Метод Ван Зиджл	0,1	0,35	0,53	0,61	0,63	0,5	0,64	0,72	0,76	0,79
Разработанный метод	0,2	0,25	0,17	0,12	0,12	0,13	0,15	0,14	0,13	0,14

4) число вершин получаемого эквивалентного базисного автомата соответствует “реальным значениям”;
5) число дуг получаемого эквивалентного базисного автомата также соответствует “реальным значениям”.

Список литературы

[1] *Melnikov B.F.* On an expansion of nondeterministic finite automata // J. of Applied Mathematics and Computing. Springer Berlin/Heidelberg. Vol. **24**. 2007. No. 1–2. P. 155–165.

[2] *Мельников Б.Ф.* Недетерминированные конечные автоматы. Тольятти: Изд-во ТГУ. – 2009. – 160 с.

[3] *Melnikov B.F., Melnikova A.A.* A new algorithm of constructing the basis finite automaton // Informatika (Lithuanian Acad. Sci. Ed.). Vol. **13**. 2002. No. 3. P. 299–310.

[4] *Melnikov B.F., Sciarini-Guryanova N.V.* Possible edges of a finite automaton defining a given regular language // The Korean J. Comp. and Appl. Math.. Vol. **9**. 2002. No. 2. P. 475–485.

[5] *Champarnaud J.M., Hansel G., Paranthoen T., Ziadi D.* Random generation models for NFA'S // J. of Automata, Languages and Combinatorics. Vol. **9**. 2004. P. 203–216.

[6] *Пивнева С.В., Рогова О.А.* Алгоритм определения репрезентативности недетерминированного конечного автомата // Электронное научное периодическое изд. "Электроника и инф. технологии" (http://fetmag.mrsu.ru/). 2009. Вып. 1.

[7] *Кнут Д.Э.* Искусство программирования. Том 2. Получисленные алгоритмы. 3-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс". 2003. 832 с.