УДК 519.6

АВТОНОМНЫЕ АВТОМАТЫ И $t(n)$–ГЕНЕРИЧЕСКИЕ ЯЗЫКИ

В.И. Донской

Таврический национальный университет,
факультет математики и информатики
ул. Ялтинская,4, Симферополь, Украина
e-mail: *donskoy@ccssu.crimea.ua*

Abstract.

The paper is dedicated to generic sets characterization in autonomous terms. In particular, it is shown that a language L is $t(n)$-generic if and only if there it no finite or infinite autonomous automationA_L generating the prefix $L|x$ of the characteristic function of the language L in the $(2^n-1)t(2^n-1)$ time. By using this result it is proved:if the language L is partially recursively enumerable and the time complexity of enumerating of all words of the language L of length at most of length $|x|=n$ for given word x is estimated as $n(2^n-1)t(2^n-1)$, then the language L is not $t(n)$- random and is not $t(n)$-v random.

1. Введение и основные определения

Автомат — это пятерка $\langle \mathcal{Q}, X, Y, G, F\rangle$, где $\mathcal{Q}, X, Y$ — соответственно внутренний, входной и выходной алфавиты. $G : \mathcal{Q}\times X \to \mathcal{Q}$ — функция переходов, $F:\mathcal{Q}\times X\to Y$ — функция выходов. Символы из множества $\mathcal{Q}$ называются *(внутренними) состояниями* автомата. Если задается некоторое выделенное (начальное) состояние $q_0\in\mathcal{Q}$, то автомат задается шестеркой $\langle \mathcal{Q}, X, Y, q_0, G, F\rangle$ и называется в этом случае *инициальным.*

Рекуррентные соотношения

$$\begin{cases} q(t) - G(q(t-1), x(t)); \\ y(t) = F(q(t), x(t)); \\ q(0) = q_0, \qquad t = 1, 2, \dots, \end{cases} \tag{1}$$

определяют *автомат второго рода* [1]. Соотношения (1) задают детерминированный оператор $T = T(\mathscr{M})$ определенный при помощи автомата $\mathscr{M}$, который преобразует последовательность входных символов $x = x(0)x(1)\dots x(t)\dots$ в выходную последовательность символов $y = y(1)y(2)\dots y(t)\dots$. Алфавиты X и Y всегда конечны. Алфавит $\mathcal{Q}$ внутрених состояний может быть конечным (в этом случае автомат $\mathscr{M}$ называется *конечным*) или бесконечным (в этом случае автомат $\mathscr{M}$ называется *бесконечным*) [1]. Определение автомата, вообще говоря, не накладывает никаких ограничений на отображения G и F, которые, следовательно, могут относиться к произвольному классу сложности.

Далее мы ограничимся использованием класса двоичных автоматов $\widetilde{M}_2^{\infty}$, у которых $X = Y = \Sigma = \{0,1\}$, а множество состояний

$$\mathcal{Q}\subseteq\{\lambda, 0, 1, 00, 01, 10, 11, \dots\} = \Sigma^*,$$

где совокупность любых двоичных строк Σ^*, для удобства упорядочена лексикографически и по длине. Во множестве $\widetilde{M_2^\infty}$ содержатся как конечные, так и бесконечные автоматы. Отображения $F : \Sigma^* \times \Sigma \to \Sigma$ и $G : \Sigma^* \times \Sigma \to \Sigma^*$ преобразуют строки двоичных символов соответственно в символы и строки. Непустое множество строк $L \subset \Sigma^*$ называется *языком, проблемой или множеством* [2]. Множество языков с зафиксированным свойством называется *классом.*

Область значений автоматного оператора $T(\mathscr{M})$, $\mathscr{M} \in \widetilde{M_2^\infty}$ будем называть *языком* $L(\mathscr{M})$ (*сверхязыком* $L^\infty(\mathscr{M})$, если автомат - бесконечный), *порождаемым автоматом* $\mathscr{M}$.

Автономным называется автомат, определяемый пятеркой $\langle \mathcal{Q}, Y, q_0, G, F\rangle$ и рекуррентными соотношениями (для автомата второго рода)

$$\begin{cases} q(t) = G(q(t-1)); \\ y(t) = F(q(t)); \\ q(0) = q_0, \qquad t = 1, 2, \dots . \end{cases} \tag{2}$$

При каждом зафиксированном начальном состоянии автономного автомата соотношения (2) однозначно определяют единственное выходное слово и *внутреннее слово* $q(0)q(1)q(2)\dots q(t)\dots$ Если автономный автомат конечен, и число его состояний равно k, то уже среди значений $\{q(0), q(1), q(2), \dots, q(k)\}$ найдутся хотя бы два одинаковых (повторяющихся) состояния. Поэтому *генерируемая автономным конечным автоматом последовательность будет* периодической с длиной периода, не превосходящей число состояний автомата.

Если $L \subset \Sigma^*$ — язык, то для произвольной строки $x \in E^*$ предикат «$x \in L$» может быть истинным (значение «1») или ложным (значение «0»). Тогда можно ввести характеристическую последовательность (строку) языка L, используя лексикографический порядок строк $x_0 = \lambda, x_1 = 0, x_2 = 1, x_3 = 00$ и т.д., и обозначить её тем же символом L с индексами $L(x_0)L(x_1)L(x_2)\dots$, причем $L(x_1) = 1$, если $x_i \in L$, и $L(x_i) = 0$ — в противном случае.

Для префикса характеристической строки длины k используется обозначение $L|x_k = L(x_0)L(x_1)\dots L(x_{k-1})$ или $L|\,y$, если $y = x_k$.

Множество называется *перечислимым*, если существует алгоритм, перечисляющий данное множество, то есть некоторая «машина» порождающая на «выходе» элементы данного множества. Предположим, указана машина, которая порождает для языка L его характеристическую последовательность $L(x_0)L(x_1)L(x_2)\dots$. Тогда язык L, очевидно, перечислим при помощи следующего алгоритма:

```
CurrentString:= x0;
For i := 0 to M step I do
    begin
        If L (CurrrentString)=1 then OutPut(CurrentString);
        CurrentSnring:=NextString(CurrentString)
```

end;

В этом алгоритме длина M начального сегмента (префикса) перечисляемой последовательности элементов языка L может быть любой. Функция *OutPut* обеспечивает «вывод» очередного элемента, а функция *NextString* преобразует любую строку x_i в следующую по лексикографическому (с учетом модификации длины строки) порядку. Для этого подсчитывается длина l текущей строки, проверяется состоящая из одних единиц, и либо формируется строка длины $l+1$, состоящая из одних нулей, либо к текущей строке как двоичному числу добавляется единица. Поэтому временная сложность для *NextString* может быть оценена как $O(|x|)$, где x - текущая строка длины $|x|$.

Далее *DTIME(t(n))* обозначается класс функций, вычислимых на машине Тьюринга с временной сложностью $O(t(n))$, и класс языков, разрешимых за время $O(t(n))$, т.е., если $L \in DTIME(t(n))$, то предикат «$x \in L$» разрешим за время $O(t(n))$.

Определение 1. $t(n)$-*предсказывающей* называется частичная функция $f : \Sigma^* \longrightarrow \{0,1\}$, такая, что $f \in DTIME(t(n))$ и *domain(f)* $\in DTIME(t(n))$. Предсказывающая функция f *плотна вдоль множества* A, если $f(A|x)$ определена для сколь угодно многих строк $x \in \Sigma^*$. Множество A *встречает (избегает) функцию* f на строке x, если $f(A|x)$ определена и $f(A|x) = A(x)\,(f(A|x) = 1 - A(x))$.

Определение 2. $t(n)$ - предсказывающая функция f называется *абсолютной* или $t(n)$ - *предикатором* для языка L , *если* L *встречает* f *на любой строке* $x \in \{0,1\}^*$.

Определение 3. Язык L называется $t(n)$ - *непредсказуемым почти всюду*, если для почти всех строк x любой $t(n)$ - предикатор f языка L требует для вычисления значения $f(L|x)$ время, большее, чем $t(n)$. В противном случае язык L называется $t(n)$ - *предсказуемым почти всюду*.

Введем обозначение $UNPRED(t(n))$ для *класса всех* $t(n)$ - *непредсказуемых почти всюду языков*. Для любого языка L из класса $UNPRED(t(n))$ не существует $t(n)$ - предикатора f, принадлежащего классу $DTIME(t(n))$ и обеспечивающего сколь угодно много правильных предсказаний вдоль L. Будем обозначать $PRED(t(n))$ — *класс всех t(n) - предсказуемых почти всюду языков.*

2. Генерические множества и предикторы

Определение 4. [2]. Условием C называется зафиксированное множество (язык) $C \subseteq \Sigma^*$. $t(n)$- условием называется условие $C \in DTIME(t(n))$. Условие C называется *плотным* вдоль множества $L \subset \Sigma^*$, если существует сколь угодно много строк x таких, что $(L|x)i \in C$ для некоторого $i \in \{0,1\}$. Множество L *встречает условие*

C, если для некоторой строки x выполняется включение $L|\,x \in C$. Множество L называется $t(n)$ - *генерическим*, если L встречает каждое $t(n)$ - условие, плотное вдоль L.

Далее класс называется $t(n)$ - генерических языков обозначается $GEN(t(n))$. Как принято в литературе [2], $t(n)$ - классы языков понимаются как $O(t(n))$ - классы. В частности, $DTIME(t(n)) = DTIME(O(t(n)))$.

Следующая теорема 1 отличается от теоремы 10 в [3] только тем, что иначе определено понятие предикатора и $t(n)$ - предсказуемых языков.

Теорема 1. Язык L является t(n)-генерическим тогда и только тогда, когда он является $t(2^n-1)$*-непредсказуемым почти всюду.*

Замечание. С использованием введенных обозначений формулировка теоремы может иметь вид $L \in GEN(t(n)) \Longleftrightarrow L \in UNPRED(t(2^n-1))$.

Доказательство. Теорема будет доказана, если убедиться, что справедливы импликации $L \notin GEN(t(n)) \Rightarrow L \notin UNPRED(t(2^n-1))$ и $L \notin UNPRED(t(2^n-1)) \Rightarrow L \notin GEN(t(n))$.

Пусть $L \notin GEN(t(n))$. Тогда существует множество $C \in DTIME(t(n))$, являющееся плотным вдоль L, такое, что язык L не встречает C. Тогда $(L|x)0 \notin C$ или $(L|x)1 \notin C$. Из условия $C \in DTIME(t(n))$ следует, что предикат «$x \in C$» разрешим за время $O(t(x))$, следовательно, существует соответствующая разрешающая функция $f_c \in DTIME(t(n))$ такая, что

$$f_c(x) = \begin{cases} 1, & x \in C; \\ 0, & x \notin C. \end{cases}$$

Определим $t(n)$-предикатор f:

$$f(L|x) = \begin{cases} f_c((L|x)0), & \textit{если } (L|x)0 \in C; \\ 1 - f_c((L|x)1), & \textit{если } (L|x)1 \in C; \\ \triangle, & \textit{иначе}, \end{cases}$$

где $\triangle$ — произвольное значение из $\{0,\ 1\}$. Поскольку L не встречает C, то $f(L|\,x) = L(x)$, если $(L|\,x)0 \in C$ или $(L|\,x)1 \in C$. Так как C — плотно вдоль L, последнее имеет место для сколь угодно многих строк, и функция $f(L|\,x)$ определена для сколь угодно многих строк x, на которых она встречает язык L. Сложность вычисления предсказывающей функции $f(L|\,x)$ определяется сложностью вычисления $f_c((L|\,x)i), i \in \{0,1\}$, и оценивается как $t(|(L|\,x)i|)$. Используя неравенство $|(L|\,x)\ | < 2^{|x|+1} - 1$ [2], легко получить оценку $|(L|\,x)i| = O(2^{|x|} - 1)$. Следовательно, указан *t(n)*-предикатор языка L для почти всех строк - функция $f \in DTIME(t(2^n-1))$. Мы полагаем всюду DTIME(*t(n)*) = DTIME(O*(t(n))*). Поэтому $L \notin UNPRED(t(2^n-1))$.

Обратно, пусть $L \notin UNPRED(t(2^n - 1))$. Тогда существует $t(n)$ - предиктор языка L — функция $f \in DTIME(t(2^{|x|} - 1))$ такая, что для сколь угодно многих строк x выполняется $f(L|x) = L(x)$. Определим условие C, как множество, состоящее из всех строк вида $(L|x)(1 - f(L|x)), \quad x \in \{0,1\}^*$. Очевидно, $C \in DTIME(t(n))$, т.к. $n = |(L|x)| > 2^{|x|} - 1$, и условие C плотно вдоль L. Но, по построению множества C, язык L не встречает условие C, следовательно, $L \notin GEN(t(n))$.

Следствие 1. Если язык L *не является* $t(n)$ - *генерическим тогда и только тогда, когда он* $t(2^n - 1)$-*предсказуем почти всюду*

$$(L \notin GEN(t(n)) \Longleftrightarrow L \notin PRED(t(2^n - 1))).$$

3. Генерические множества и автономные автоматы

Теорема 2. Следующие высказывания эквивалентны.

1. Для языка L существует $t(2^n - 1)$ - предикатор.
2. Существует бесконечный автономный автомат A_L, порождающий префикс $L|x$ характеристической функции языка L за время $(2^n - 1)t(2^n - 1)$.

Доказательство. Пусть f_L — $t(2^n - 1)$-предиктор для языка L, и $L(x_0)L(x_1)L(x_2)\ldots$ — характеристическая последовательность языка L в соответствии с лексикографическим порядком $x_0 = \lambda, \quad x_1 = 0, \quad x_2 = 1, \quad x_3 = 00, \quad x_4 = 01\ldots.$

Пометим состояния автомата двоичными строками $q_0 = \lambda, \quad q_1 = L(x_0), \quad q_2 = L(x_0)L(x_1), \ldots.$ Функцию G переходов автомата A_L определим как $G(q(\tau - 1)) = 2 \cdot q(\tau - 1) + f_L(L|\, x_\tau)$, где $\tau = 1, 2, \ldots$ – шаги вычислений; умножение двоичной строки на 2 соответствует ее сдвигу влево на один разряд (умножение на 2 пустой строки дает по определению пустую строку); добавление двоичного символа в младший разряд обеспечивается предиктором. Функцию выходов F автомата A_L определим как $F(q(\tau)) = f_L(L|\, q(\tau))$. Полученный автономный автомат второго рода порождает префикс $L|\, x$, выполняя $\tau \leqslant |L|\, x| = O(2^{|x|} - 1)$ шагов, гарантированно вычисляя на каждом шаге предиктор за время $O(t(2^{|x|} - 1))$, откуда следует оценка сложности $(2^n - 1)t(2^n - 1)$.

Обратно, если существует автономный бесконечный автомат A_L, порождающий префикс $L|\, x$ характеристической функции языка L за время $(2^n - 1)t(2^n - 1)$, то он, генерируя префикс $L|\, x$, выполняет $2^n - 1$ шагов. Осуществляя перекодировку состояний автомата A_l строками, — последовательно генерируемыми префиксами языка L, — можно получить изоморфный автомат, функция выходов которого и будет абсолютным $t(2^n - 1)$-предиктором.

Следствие 2. *Язык L является $t(n)$-генерическим тогда и только тогда, когда не существует конечный или бесконечный автономный автомат A_L, порождающий префикс $L|\,x$ характеристической функции языка L за время $(2^n-1)t(2^n-1)$.*

4. Перечислимость и случайность

Рассматриваемый в статье автономно-автоматный подход к изучению алгоритмической случайности перспективен для практических приложений и может быть использован для обоснования найденных закономерностей в данных путем расшифровки автоматов. Кроме этого он позволяет получить ряд теоретических результатов, в частности, показанных ниже. Некоторые достоинства автономно-автоматного подхода видны при сравнении полученных результатов с результатами работы [2], основанными на характеризации генерических множеств при помощи машин Тьюринга с оракулом.

Следствие 3. *Язык L не является $t(n)$-генерическим тогда и только тогда, когда он частично-рекурсивно перечислим и временная сложность перечисления всех слов языка L, не превышающих заданное слово x длины $|\,x|=n$, оценивается как $n(2^n-1)t(2^n-1)$.*

Определение 5. [2, 3, 4, 5] *Мартингалом* называется функция $d:\Sigma^*\to[0,\infty)$ такая, что для любой строки $\omega\in\Sigma^*$ выполняется соотношение $d(\omega 0)+d(\omega 1)\leq 2d(\omega)$. Мартингал d называется успешным для проблемы $L\subset\Sigma^*$, если $\limsup_n d(L|z_n)=\infty$, где $z_0,z_1,z_2,\ldots,z_n,\ldots$ — строки из Σ^* в лексикографическом порядке. Мартингал d называется $t(n)$-*вычислимым* или $t(n)$-*мартингалом*, если существует функция $\hat{d}:N\times\Sigma^*\to\mathcal{Q}$ такая, что $\hat{d}\in DTIME(t(n))$ и для любого натурального числа r и любой строки $\omega\in\Sigma^*$ выполняется неравенство $|\hat{d}_r(\omega)-d(\omega)|\leqslant 2^{-r}$, где N и Q, соответственно, множества натуральных и рациональных чисел.

Определение 6. [2, 3, 4, 5] Множество $L\subset\Sigma^*$ называется $t(n)$- *случайным*, если для любого $t(n)$- мартингала $d:\Sigma^*\to[0,\infty)$ выполняется $\limsup_n d(L|z_n)<\infty$, т.е. не найдется $t(n)$- мартингала, успешного на L.

Класс всех $t(n)$-случайных множеств(языков) обозначается $RAND(t(n))$.

Установлено [1, 4],что каждый $t(n)$-случайный язык является $t(n)$-генерическим, то есть

$$RAND(t(n))\subset GEN(t(n)). \tag{3}$$

Аналогичный результат для $t(n)$-ν-случайных языков над любой строго положительной $t(n)$- вычислимой мерой ν получен в [4]:

$$RAND_\nu(t(n))\subset GEN(t(n)). \tag{4}$$

Теорема 3. *Если язык L частично-рекурсивно перечислим и временная сложность перечисления всех слов языка L, не превышающих заданное слово x длины $|x| = n$, оценивается как $n(2^n - 1)t(2^n - 1)$,то язык L не является $t(n)$-случайным и не является $t(n)$-ν- случайным.*

Доказательство. Доказательство следует из соотношений (3), (4) и следствия 3. Действительно, $L \notin GEN(t(n)) \Longrightarrow L \notin RAND(t(n))$ и $L \notin GEN(t(n)) \Longrightarrow L \notin RAND_\nu(t(n))$, а условие $L \notin GEN(t(n))$ в силу следствия 3 равносильно выполнению достаточного условия теоремы.

Список литературы

1. Трахтенброт Б.А., Бардзинь Я.М. Конечные автоматы (поведение и синтез) М. :Наука,1970. - 400с.
2. Ambos-Spies K., Mayordomo E., Wang Y., Zheng X. Resource-Bounded Balanced Genericity, Stochasticity and Weak Randomness//In Proc. of the 13^{th} Symp. of Comp. Sci. Springer-Verlag, 1996, p. 63-74.
3. Pavan A. and Selman A. L. Separation of NP - completeness Notions. Dept. of CS&Eng. Univ. at Buffalo, NY 14260, December 6, 2000, p. 1 - 17.
4. Lorentz A.K. and Lutz J.H. Genericity and Randomness over Feasible Probability Measures//Theoretical Comp. Sci., 1998, Vol. 207, N1, p. 245-259.
5. Ambos-Spies K. and Mayordomo E. Resource-Bounded Measure and Randomness//In. A. Sorbi, editor, Complexity Logic and Recursion Theory. Lecture Notes In Pure and Applied Math. Marcel Dekker, New York, 1996, p. 1-47.