Компьютерные инструменты в образовании, 2014
№ 6: 23–27
УДК: 519.682.1 + 681.142.2
http://ipo.spb.ru/journal

О МЕТОДЕ МИНИМИЗАЦИИ КОНЕЧНЫХ АВТОМАТОВ

Мартыненко Борис Константинович

Аннотация

Метод минимизации приведённых детерминированных конечных автоматов, описанный в частности в [1], не всегда даёт минимальный по числу состояний автомат. В данной статье предлагается процесс минимизации этим методом считать лишь первым этапом и дополнить его вторым этапом, гарантирующим искомый результат.

Ключевые слова: *конечный автомат, минимизация.*

1. ВВЕДЕНИЕ

Прежде всего, дадим несколько необходимых определений.

Определение 1. *Детерминированный конечный автомат* (dfa) M есть пятерка

$$M = (Q, \Sigma, \delta, q_0, F),$$

где Q — конечное множество состояний;

Σ — входной алфавит;

$\delta : Q \times \Sigma \to Q$ — функция изменения состояния;

$q_0 \in Q$ — стартовое состояние;

$F \subseteq Q$ — множество конечных состояний.

Функция перехода δ называется *полной*, если она определена для каждой пары в $Q \times \Sigma$.

Если δ является полной, то мы называем M *полным* dfa.

Если представить dfa как ориентированный граф, где состояния — это узлы, и переходы между состояниями — дуги между узлами, то в вышеупомянутом определении не требуется, чтобы dfa был *связаным* [2], то есть не требуется, чтобы всегда существовал ориентированный маршрут, соединяющий любую пару его различных вершин. На рис. 1 представлен dfa в виде диаграммы переходов, который является связаным полным графом из одной компоненты.

Конечный автомат, такой что каждое состояние достижимо из стартового состояния и достигает конечного состояния, называют *приведёнными* dfa. Приведённый dfa может быть не полным dfa.

Следуя [1], определим отношение $\equiv$ *неразличимости* на множестве состояний приведённого детерминированного конечного автомата.

Определение 2. Пусть $M = (Q, \Sigma, \delta, q_0, F)$ — приведённый dfa с n состояниями.

Пусть $q_1, q_2 \in Q$. Считается, что $q_1 \equiv^k q_2$ при двух следующих условиях:

© Мартыненко Б.К., 2014

(1) $q_1 \equiv^0 q_2$ *тогда и только тогда, когда* q_1 *и* q_2 *оба либо принадлежат, либо не принадлежат* F,

(2) $q_1 \equiv^k q_2$ *тогда и только тогда, когда* $q_1 \equiv^{k-1} q_2$ *и* $\delta(q_1, a) \equiv^{k-1} \delta(q_2, a)$ для всех $a \in \Sigma$.

По лемме 2.11 из [1] (*см.* 2.3: с. 149) состояния q_1 и q_2 неразличимы тогда и только тогда, когда они $(n-2)$-не различимы.

Отношение $\equiv^0$ грубейшее; оно разбивает Q на два класса: F и Q/F. Если $\equiv^{k+1} \neq \equiv^k$, то отношение $\equiv^{k+1}$ тоньше, чем $\equiv^k$, то есть в нём, по крайней мере, на один класс эквивалентности больше, чем в $\equiv^k$. Так как каждое из множеств F и Q/F содержит не более $n-1$ элементов, можно получить не более $n-2$ последовательных утончений отношения $\equiv^0$.

Если $\equiv^{k+1} = \equiv^k$ для некоторого k, то, в силу условия (2), $\equiv^{k+1} = \equiv^{k+2} = \ldots$. Таким образом, $\equiv$ — это первое из отношений $\equiv^k$, для которых $\equiv^{k+1} = \equiv^k$.

Приведём пример, показывающий, что метод минимизации, о котором идёт речь, основанный на построении сходящейся последовательности отношений $\equiv^k$ для $k = 0, 1, 2, \ldots$, *не всегда* даёт минимальный по числу состояний конечный автомат.

На этом же примере покажем, как дополнить процесс получения минимального автомата, эквивалентного исходному автомату.

2. КОНТРПРИМЕР

Следуя [3: Ch. 2, p. 45–46], пусть A_0 — автомат, который читает цепочки из 0 и 1 и распознаёт эти цепочки как двоичные числа, конгруэнтные 2 (по модулю 3). Мы используем $v_3(x)$, чтобы обозначить значение бинарной цепочки x по модулю 3. Например, $v_3(100) = 1$ и $v_3(1011) = 2$.

Рассмотрим произвольную входную цепочку $w = a_1 \ldots a_n$ в A_0, где каждый a_i, $1 \le i \le n$, есть или 0 или 1. Ясно, что для каждого i, $1 \le i \le n$, цепочка $a_1 \ldots a_i$ попадает в один из этих трёх случаев: (0) $v_3(a_1 \ldots a_i) = 0$, (1) $v_3(a_1 \ldots a_i) = 1$ и (2) $v_3(a_1 \ldots a_i) = 2$. Никакие другие случаи невозможны. Так что A_0 нуждается только в трёх состояниях, которые соответствуют вышеупомянутым трём случаям (и начальное состояние соответствует случаю (0)). Обозначим эти три состояния (0), (1), и (2), соответственно. Правила, которые управляют изменениями состояний, должны быть определены соответственно.

Заметим, что

$$v_3(a_1 \ldots a_{i+1}) \equiv 2 * v_3(a_1 \ldots a_i) + a_{i+1} \pmod 3.$$

Так, если текущее состояние есть (1) и текущий входной символ есть 1, то следующее состояние есть (0), поскольку $2 * 1 + 1 = 0 \pmod 3$. Состояния и их правила перехода показаны на рис. 1.

Ясно, что каждый шаг изменения состояния единственным образом определён текущим состоянием и текущим входным символом. Мы выделяем состояние (2) как конечное состояние и определяем, что A_0 принимает вход w, если A_0 находится в состоянии (2) после чтения последнего символа w.

Итак, имеем dfa $A_0 = (Q, \Sigma, \delta, q_0, F)$,

где $Q = \{p_0, p_1, p_2\}$,

$\Sigma = \{0, 1\}$, $F = \{p_2\}$, $q_0 = p_0$,

$\delta(p_0, 0) = p_0$, $\delta(p_0, 1) = p_1$,

$\delta(p_1, 0) = p_2$, $\delta(p_1, 1) = p_0$,

$\delta(p_2, 0) = p_1$, $\delta(p_2, 1) = p_2$.

A_0 — пример полного детерминированного конечного автомата (dfa), причём *минимального* по числу состояний. Его диаграмма переходов представлена на рис. 1.

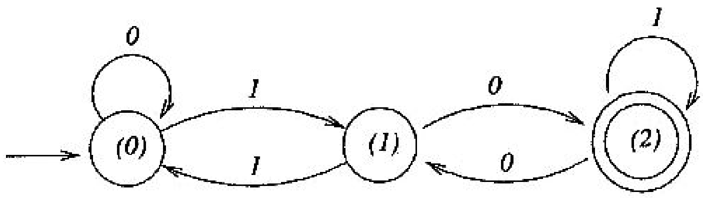

Рис. 1. Диаграмма переходов dfa A_0

Судя по диаграмме переходов, A_0 — приведённый детерминированный конечный автомат, распознающий язык, определяемый регулярным выражением $0^*(11)^*1(00)^*01^*$.

По этой же диаграмме переходов можно легко построить dfa M и соответствующую матрицу переходов (табл. 1). Результат представлен на рис. 2.

Дан dfa $M = (Q, \Sigma, \delta, q_0, F)$,
где $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$,
$\Sigma = \{0, 1\}$,
$F = \{q_2, q_5\}$,
$\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$,
$\delta(q_1, 0) = q_2$, $\delta(q_1, 1) = q_3$,
$\delta(q_2, 0) = q_4$, $\delta(q_2, 1) = q_5$,
$\delta(q_3, 0) = A$, $\delta(q_3, 1) = q_1$,
$\delta(q_4, 0) = q_2$, $\delta(q_4, 1) = A$,
$\delta(q_5, 0) = A$, $\delta(q_5, 1) = q_5$.

Табл. 1

δ	**0**	**1**	ε
0	0	1	
1	2	3	
2	4	5	+
3	*A*	1	
4	2	*A*	
5	*A*	5	+

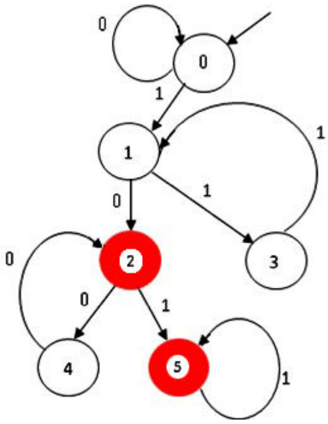

Рис. 2. Три равнозначных представления dfa M

В определении отображения δ и в табл. 1 символ A обозначает, что движение автомата в соответствующих состояниях и входных символов не определено. Следуя [1], разобьём всё множество состояний данного автомата на два подмножества:
$\equiv^0$: {2, 5} и {0, 1, 3, 4 , A}.

Дальнейший процесс исключает по одному состоянию из этих двух подмножеств и образующихся подобных подмножеств по признаку различимости пар состояний.

Действия автомата в состоянии 0: $\to^0 0 \;\; \to^1 1$.

Так как переходные состояния 1 и 0 в одном и том же подмножестве, то отношение $\equiv^0$ не изменяется.

Действия автомата в состоянии 1: $\to^0 2 \;\; \to^1 3$.

Так как переходные состояния 2 и 3 в разных подмножествах, то отношение $\equiv^0$ изменяется на следующее:
$\equiv^1$: {2, 5}, {1}, {0, 3, 4, A}.

Действия автомата в состоянии 2: $\to^0 4 \;\; \to^1 5$.

Так как переходные состояния 4 и 5 в разных подмножествах, то отношение $\equiv^1$ изменяется на следующее:
$\equiv^2$: {2}, {5}, {1}, {0, 3, 4, A}.

Действия автомата в состоянии 3: $\to^0 A \;\; \to^1 1$.

Так как переходные состояния A и 1 в разных подмножествах, то отношение $\equiv^2$ изменяется на следующее:
$\equiv^3$: {2}, {5}, {1}, {3}, {0, 4, A}.

Действия автомата в состоянии 4: $\to^0 2 \quad \to^1 A$.

Так как переходные состояния 2 и A в разных подмножествах, то отношение $\equiv^3$ изменяется на следующее:

$\equiv^4$: $\{2\}, \{5\}, \{1\}, \{3\}, \{4\}, \{0, A\}$.

Действия автомата в состоянии 5: $\to^0 A \quad \to^1 5$.

Это не добавляет нечего нового в отношение $\equiv^4$. Процесс закончен, и отношение $\equiv = \equiv^4$. Результат минимизации $M' = M$.

Например, цепочка 1110001 принимается dfa M':

$q_0 \to^1 q_1 \to^1 q_3 \to^1 q_1 \to^0 q_2 \to^0 q_4 \to^0 q_2 \to^1 q_5$.

Итак, по методу [1] (*см.* гл. 2, теор. 2.6, стр. 150) данный автомат *минимальный*.

Однако это не так, ибо dfa A_0 распознаёт тот же самый язык $0^*(11)^*1(00)^*01^*$, но имеет всего три состояния.

3. ДООПРЕДЕЛЕНИЕ МЕТОДА МИНИМИЗАЦИИ КОНЕЧНЫХ АВТОМАТОВ

Чтобы исправить метод минимизации приведённых детерминированных конечных автоматов, описанный в [1], достаточно считать его первым этапом процесса.

Второй этап процесса минимизации автомата базируется на следующих соображениях.

Перебираются все пары $(q_1, q_2) \in Q \times Q$, $q_1 \neq q_2$, полученные после первого этапа минимизации по [1]. Пусть q_1 и q_2 — два различных состояния автомата M'. Соответствующие строки таблицы переходов окрашиваются в один и тот же цвет, если и только если для *всех* входных символов $a \in \Sigma \cup \{\varepsilon\}$ значения $\delta(q_1, a)$ и $\delta(q_2, a)$ оба не определены или определено только одно из этих значений, либо $\delta(q_1, a) = \delta(q_2, a)$.

В применении к нашему примеру результат раскраски табл. 1 представлен на табл. 1*а*.

Табл. 1*а*

δ	0	1	ε
0	0	1	
1	2	3	
2	4	5	+
3		1	
4	2		
5		5	+

Табл. 1*б*

δ	0	1	ε
0–3	0–3	1–4	
1–4	2–5	0–3	
2-5	1–4	2–5	+
3		1	
4	2		
5		5	+

Исключить!

Табл. 2

δ	0	1	ε
p_0	p_0	p_1	
p_1	p_2	p_0	
p_2	p_1	p_2	+

Сопоставляя строчки таблицы переходов табл. 1*а* этого автомата, можем заметить, что строчки, помеченные одним цветом, а именно, 0 и 3, 1 и 4, 2 и 5, можно слить в одну, 0–3, 1–4, 2–5, соответственно перенумеровать переходные состояния и исключить дубликаты строк 3, 4, 5 (*см.* табл. 1*б*). Окончательный результат минимизации представлен dfa M'' в табл. 2.

Итак, dfa M'', полученный в два этапа, равен dfa A_0.

4. ЗАКЛЮЧЕНИЕ

Теорема 2.6 из [1, с. 150], утверждающая, что

Автомат M', который строится алгоритмом 2.2, имеет наименьшее число состояний среди всех конечных автоматов, допускающих язык $L(M)$,

не верна.

Достаточно в этой формулировке заменить M' на M'', чтобы утверждение стало верно.

Вопросы. При каких ограничениях на dfa даёт минимальный автомат:

1) метод, описанный в [1];

2) приём, использующий дополнительный этап минимизации, применённый в контерпримере.

Проблема. Построить корректный метод минимизации приведённого dfa при допущении, что его диаграмма переходов не является *связным графом*.

Литература

1. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ. М.: Мир, 1978.
2. *Харари Ф.* Теория графов. Второе издание. УРСС, М., 2003.
3. *Rozenberg G., Salomaa A.* Handbook of Formal Languages. Vol. 1: Word, Language, Grammar. Berlin, Heidelberg: Springer-Verlag, 1997.

ABOUT THE METHOD FOR MINIMIZATION OF FINITE AUTOMATA

Martynenko B. K.

Abstract

The method for minimizing reduced deterministic finite automata, described in particular in [1], not always gives minimum on number of it states. It is offered to add the second stage to the method in question in order to guarantee required outcome.

Keywords: *finite automaton, minimization.*

Мартыненко Борис Константинович,
доктор физико-математических наук,
профессор кафедры информатики
математико-механического факультета
СПбГУ,

mbk@ctinet.ru

© Наши авторы, 2014.
Our authors, 2014.