

УДК 519.713

Ю.К. Апраксин, профессор, д-р техн. наук,

А.П. Зезюлинский,

А.А. Сахнов

Севастопольский национальный технический университет

ул. Университетская 33, г. Севастополь, Украина, 99053

E-mail: kvt@sevgtu.sebastopol.ua

ИССЛЕДОВАНИЕ СПОСОБОВ ПРЕДСТАВЛЕНИЯ ДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА

Исследуются различные способы представления детерминированного конечного автомата согласно критериям оптимальности: времени поиска перехода и затрат памяти на представление модели автомата.

Ключевые слова: конечный автомат, модель, способ представления, бинарный поиск.

Модель конечного автомата широко используется при проектировании аппаратного, программного, системного программного обеспечения вычислительных систем различного назначения (компиляторов на фазах лексического и синтаксического анализа, сетевых протоколов, устройств управления и т.д.) [1].

Математическое представление конечного автомата (КА):

$$KA = \langle X, Y, S, S_0, \varphi, \Psi \rangle,$$

где X, Y, S – соответственно конечные множества входов, выходов, состояний, S_0 ($S_0 \in S$) – начальное состояние, φ, Ψ – функции переходов и выходов.

Если для каждой пары (x, s) , где $x \in X, s \in S$ определен переход в какое-либо состояние $s' \in S$ – говорят о полностью определенном конечном автомате. В противном случае конечный автомат частично определен. Детерминированный конечный автомат имеет для любого состояния не более одного перехода по одному символу входного алфавита.

Функция выходов $\Psi(x, s) = 1$, если $\varphi(x, s) = s'$, где $s' \in F$, где F ($F \subset S$) – множество конечных состояний.

Суть функционирования конечного автомата заключается в преобразовании последовательности входов в последовательность выходов. Обладая таким преимуществом как высокий уровень абстрактности [2], конечно-автоматная модель имеет и существенный недостаток: в случае линейного повышения сложности задачи управления экспоненциально растет число состояний автомата, и, как следствие, растут затраты памяти на реализацию модели и время поиска перехода.

Целью работы является разработка программной системы формирования машинного представления детерминированного конечного автомата в зависимости от числа состояний и насыщенности автомата переходами. Критериями оптимальности представления детерминированного конечного автомата являются: стоимостные затраты (общий объем памяти, необходимый для представления модели детерминированного конечного автомата в ЭВМ), временные затраты (среднее время поиска произвольного перехода).

Рассмотрению подлежат детерминированные частично определенные конечные автоматы. В соответствии с [3] рассматриваются следующие способы представления конечного автомата: таблица переходов, список переходов, граф переходов.

Таблица переходов (рисунок 1) реализована в виде одномерного массива. Размерность таблицы переходов W определяется мощностью входного алфавита $|X|$ и числом состояний автомата $|S|$: $W = |X| \cdot |S|$. Элемент таблицы переходов – номер состояния перехода.

В случае малой насыщенности переходов большая часть таблицы не определена переходами, что крайне избыточно и приводит к значительным затратам памяти. Под насыщенностью конечного автомата будем понимать отношение числа определенных переходов к числу всех возможных переходов.

Из рисунка 1 видно, что учет только определенных переходов в таблице переходов КА позволит избежать избыточности и, как следствие, снизить затраты памяти на представление модели автомата в ЭВМ. Рассмотрим представления автомата, содержащие только определенные переходы: список эквивалентных переходов, список переходов, граф переходов.

		Состояния												
$X_i \backslash S_j$		0	1	2	3	4	5	6	7	...	N			
Входной алфавит	'1'	1								...		N — Конечное состояние 0 — Начальное состояние — Неопределенный переход		
	...	...											...	
	'a'	2		3		5	5			...	7			
	'b'	2		3			5							
	'c'	2		4	5		6							
	'd'	2	3		1		0							
	'e'													
	'f'							7						
	'g'		1						7	...				
	'h'										0			
...	...										...			
'q'		4						5		...				

Рисунок 1 — Реализация таблицы переходов

Список эквивалентных переходов (рисунок 2) состоит из таблицы эквивалентных переходов и таблицы сегментов (два одномерных массива). Элемент таблицы эквивалентных переходов есть пара — (число повторений, состояние перехода). Таблица сегментов индексирует переходы для каждого из состояний автомата. Элемент таблицы сегментов — индекс в таблице эквивалентных переходов. Список эквивалентных переходов позволяет значительно сократить затраты памяти на представление, но для поиска перехода требуется последовательно разворачивать таблицу эквивалентных переходов, что значительно замедляет быстрдействие поиска. Представление автомата в виде списка эквивалентных переходов — альтернатива k-эквивалентным разбиениям [3], назначение которых заключается в минимизации числа состояний конечного автомата путем выделения эквивалентных классов состояний.

Таблица эквивалентных переходов		Таблица сегментов	
Индекс таблицы	0 1 2 3 4 5 6 7 8	0 1	3 9
номер состояния	1 2 5 4 3 2 1 1 4	Граничный индекс в таблице эквивалентных переходов	
число повторений	1 4 5 4 1 2 1 1 1	Индекс (Номер состояния)	
		0 состоянию соответствуют переходы: от 0 до 3 в таблице эквивалентных переходов	
		1 состоянию соответствуют переходы: от 3 до 9 в таблице эквивалентных переходов	

Рисунок 2 — Реализация списка эквивалентных переходов

Список переходов (рисунок 3) состоит из таблицы переходов для состояний и таблицы сегментов (два одномерных массива). Таблица переходов для состояний содержит переходы для всех состояний.

Список переходов		Таблица сегментов	
Индекс таблицы	0 1 2 3 4 5 6 7	0 1	5 8
символ входного алфавита	'1' 'a' 'b' 'c' 'd' 'd' 'g' 'q'	Граничный индекс в списке переходов	
переход	1 2 2 2 5 3 1 4	Индекс (Номер состояния)	
		0 состоянию соответствуют переходы: от 0 до 5 в списке переходов	
		1 состоянию соответствуют переходы: от 5 до 8 в списке переходов	

Рисунок 3 — Реализация списка переходов

Элемент таблицы переходов есть пара — (символ входного алфавита, переход). Таблица сегментов индексирует таблицу переходов для каждого из состояний автомата. Элемент таблицы сегментов — индекс таблицы переходов. Переходы отсортированы по символу перехода, это позволяет применять алгоритм бинарного поиска перехода. Таблица переходов содержит только определенные переходы. Двоичный поиск и хранение только существующих переходов позволяют повысить быстродействие поиска и снизить затраты памяти на представление модели в ЭВМ.

Граф переходов (рисунок 4) реализован массивом объектов. Каждый объект — состояние, которое содержит таблицы переходов. Объекты не хранят переходов в инициализирующее состояние. Элемент таблицы переходов для состояния представлен как (входной символ; переход). В связи с тем, что переходы отсортированы по входному символу, возможно применение бинарного поиска перехода. Хранение только возможных переходов позволяет снизить затраты памяти, но на хранение объектов состояний требуются дополнительные расходы памяти.

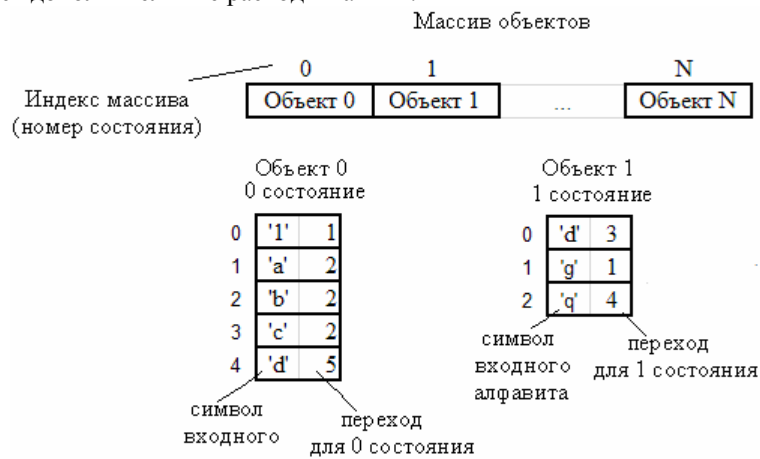


Рисунок 4 — Реализация графа переходов

Для анализа способов представления конечного автомата разработана программная система, позволяющая оценить затраты памяти на реализацию представления, время поиска перехода для автомата с заданными числом состояний, насыщенностью переходами, мощностью входного алфавита, начальным состоянием. Эксперимент проводился для $|S| \leq 2^{16}$.

В ходе исследования получены характеристики затрат памяти и времени поиска для списка эквивалентных переходов, списка переходов, графа переходов в сравнении с таблицей переходов.

На рисунке 5 изображена зависимость параметра памяти M_p , характеризующего во сколько раз затраты памяти на ЭВМ на рассматриваемые представления детерминированного конечного автомата меньше, чем на представление с помощью таблицы переходов. Чем выше параметр памяти, тем эффективнее исследуемое представление по критерию затрат памяти.

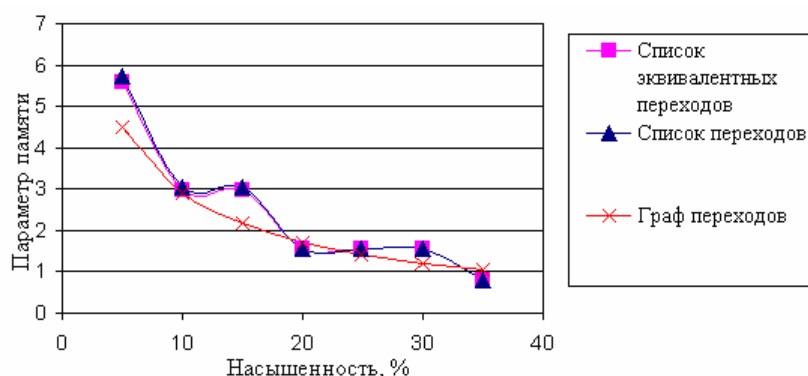


Рисунок 5 — Зависимость показателя памяти представления от насыщенности автомата переходами

Параметр времени P_p (рисунок 6) характеризует во сколько раз время поиска перехода для рассматриваемых представлений медленнее, чем время поиска перехода для представления с помощью таблицы переходов. Чем ниже параметр времени, тем эффективнее рассматриваемое представление по критерию времени поиска перехода.

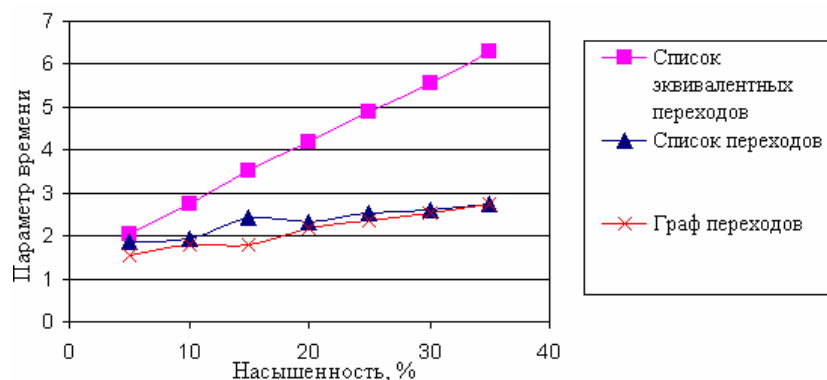


Рисунок 6 — Зависимость показателя времени поиска перехода от насыщенности автомата переходами

Введена оценочная функция O_p способа представления детерминированного конечного автомата $O_p = \frac{k \cdot M_p}{P_p}$, $k \geq 1$, где k — коэффициент приоритета, который характеризует приоритет параметра памяти над параметром времени.

График зависимости оценочной функции от насыщенности автомата переходами показан на рисунке 7.

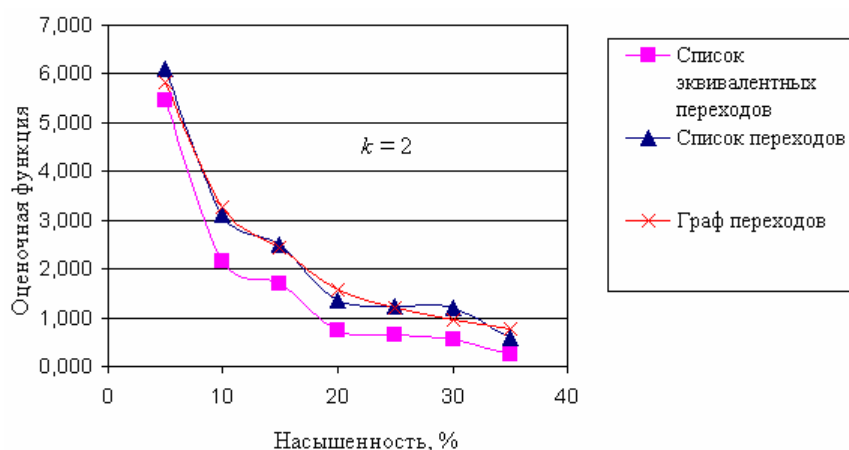


Рисунок 7 - График зависимости оценочной функции представления от насыщенности автомата

В результате тестирования представлений автоматов на множестве автоматных моделей с различной насыщенностью найдено, что если насыщенности автомата переходами менее 30 %, то оптимальными представлениями детерминированного частично определенного конечного автомата являются список переходов или граф переходов.

В качестве оптимального способа представления детерминированного частично определенного конечного автомата выбрано представление с помощью списка переходов. При использовании списка переходов в среднем экономия памяти на представление автоматной модели на ЭВМ составляет 32,8 %, рост производительности операции поиска перехода — 3,48 %.

В дальнейшем предполагается исследовать возможности применения разработанной системы для автоматов с мощностью состояний существенно превышающей 2^{16} (для суперсложных систем управления).

Библиографический список

1. Карпов Ю.Г. Теория автоматов / Ю.Г. Карпов. — СПб: Питер, 2002. — 224 с.
2. Мелихов А.Н. Ориентированные графы и автоматы. / А.Н. Мелихов. — М.: Наука, 1971. — 415 с.
3. Гилл А. Введение в теорию конечных автоматов./ А. Гилл. — М.: Наука, 1966. — 272 с.

Поступила в редакцию 01.06.2009 г.