УДК 519.1: 681.3

В.Е.ХАЧАТРЯН, Я.Г.ВЕЛИКАЯ, А.И.СУНЦОВА

СТРУКТУРНЫЕ ПРЕОБРАЗОВАНИЯ МОДЕЛИ ВЫЧИСЛЕНИЙ

Для распознавания эквивалентности некоторых моделей вычислений применим трансформационный метод. Последний использует графическое представление модели, для которой строится древовидное покрытие. В тех случаях, когда трансформационный метод позволяет доказать разрешимость проблемы эквивалентности, древовидное покрытие строится однозначно. В общем случае построение такого покрытия осуществляется неоднозначно. В работе предлагается такое преобразование модели, после которого измененная модель, оставаясь в классе эквивалентности, обладает единственным покрытием.

Ключевые слова: *модель вычислений; проблема эквивалентности; трансформационный метод.*

For recognition of equivalence of some models of calculations it is applicable a transformation method. Last uses graphic representation of model for which the treelike covering is under construction. When the transformation method allows to prove resolvability of a problem of equivalence a treelike covering is under construction unequivocally. Generally construction of such covering is carried out ambiguously. In work such transformation of model after which the changed model, remaining in an equivalence class is offered, possesses a unique covering.

Keywords: *model of calculations; equivalence problem; transformation method.*

Под моделью вычислений в широком смысле понимают множество конструктивных объектов с приписанной ему универсальной процедурой, посредством которой каждому объекту сопоставляется порожденное ему множество.

Одной из фундаментальных проблем моделей вычислений является проблема эквивалентности – нахождение алгоритма, который по любой паре объектов модели определяет, обладают они одинаковыми порождаемыми множествами или нет.

В работах [1,2,3,4,5] рассматривается трансформационный метод распознавания эквивалентности, который для некоторых моделей вычислений, в частности, для многоленточных автоматов с непересекающимися циклами [2], задает для проблемы эквивалентности разрешающий алгоритм. Рассмотрим использование трансформационного метода для двухленточных бинарных автоматов.

Двухленточный бинарный автомат будем представлять в виде диаграммы переходов, которая представляет собой конечный ориентированный граф с размеченными вершинами и дугами.

Диаграмма переходов двухленточного бинарного автомата определена над алфавитами $P=\{p,q\}$ и $Q=\{0,1\}$ и удовлетворяет следующим свойствам:

– вершины графа помечены символами алфавита P, а дуги – символами алфавита Q;

– имеются две выделенные вершины: вход и выход автомата; из выхода нет исходящих дуг, а из других вершин исходит по две дуги, причем из каждой вершины исходят дуги, помеченные разными символами.

Введем понятия, используемые при определении эквивалентности рассматриваемых автоматов.

Пусть D – диаграмма над P и Q. Конечный ориентированный путь w в диаграмме D задается последовательностью дуг. Он описывается историей $L(w)$, где $L(w) = (a_1, e_1)(a_2, e_2)\ldots(a_k, e_k)$, a_j – метка вершины, из которой исходит j-ая дуга пути, а ε_j – метка этой дуги, j = 1, 2, …, k. p-проекцией (соответственно, q-проекцией) пути w называется слово, полученное из $L(w)$ удалением всех пар, не содержащих символа p, (соответственно, символа q).

Путь w, начинающийся во входе диаграммы, назовем маршрутом; если последний оканчивается в выходе, то маршрутом через диаграмму. Автоматы D_1, D_2, по определению, *эквивалентны* тогда и только тогда, когда для любого маршрута через D_j , j = 1, 2 существует маршрут через D_{3-j} , p-проекция и q-проекция которого, совпадают, соответственно, с p-проекцией и q-проекцией первого маршрута.

При таком задании многоленточного автомата символ, сопоставляемый вершине, обозначает ленту, с которой автомат работает в состоянии, идентифицируемом этой вершиной, а метка дуги, выходящей из вершины – символ, расположенный на этой ленте и определяющий выбор следующего состояния автомата.

Автоматы D_1, D_2 назовем *сильно эквивалентными* тогда и только тогда, когда множества историй всех путей через эти диаграммы совпадают. Отметим, что этот случай эквивалентности можно трактовать как обычную эквивалентность детерминированных конечных автоматов.

При изображении диаграммы условимся опускать состояния вместе с приходящими в них дугами, не принадлежащие маршрутам через диаграмму. Примеры сильно эквивалентных автоматов D_1 и D_2 и эквивалентных автоматов D_3 и D_4 над алфавитами $P=\{p,q\}$ и $Q=\{0,1\}$ приведены на рисунке 1; дуги, которые должны нести символ 1, снабжены жирной точкой у их основания, что позволяет игнорировать метки дуг; вход изображен черным кружком, а выход – перечеркнутым кружком.

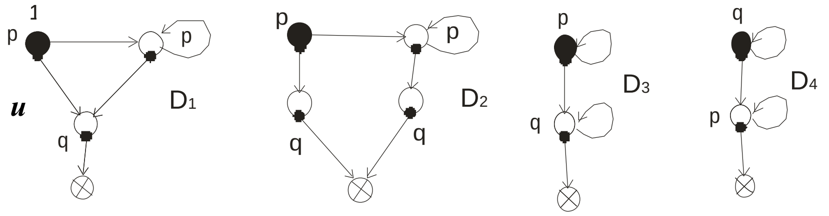

Рисунок 1 – Примеры сильно эквивалентных автоматов D_1 и D_2 эквивалентных автоматов D_3 и D_4 над алфавитами $P=\{p,q\}$ и $Q=\{0,1\}$

Определим характеристику автомата D, называемую *покрытием*. Это древовидный фрагмент F_D, все вершины и дуги которого являются образами вершин и дуг автомата D с их метками. Корнем является образ входа автомата D. Обозначим через V список всех вершин автомата D, лежащих на ее маршрутах через автомат, за исключением ее выхода. Внося в F_D какую-либо вершину из списка, будем вычеркивать ее образ из V.

На первом шаге в F_D вносится корень – образ входа v_0 автомата D, и вершина v_0 удаляется из списка V. Пусть на некотором шаге в F_D внесена вершина u, являющаяся образом вершины v автомата D, и вершина v вычеркнута из V, причём вершина u – не выход и из неё ещё не выходят дуги. Обозначим α_1 и α_2 – дуги, исходящие из вершины v. Пусть α_I, i=1,2 оканчивается в вершине v_i . Если v_i содержится в V, создаем образ вершины v_i и направляем в нее дугу α_i; удаляем v_i из списка V. Если v_i не содержится в списке V, но содержалась ранее и не является выходом, то создаем образ вершины v_i; объявляем его выходом фрагмента F_D и в нее направляем дугу α_i с ее меткой.

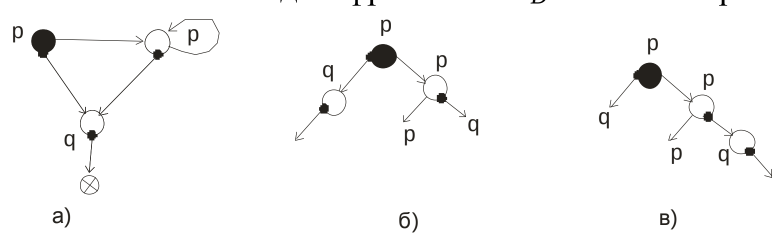

Рисунок 2 – Примеры покрытий для автоматов

Если v_i не содержится в списке V и не содержалась там, то строящийся фрагмент F_D не меняется. Наконец, если v_i

является выходом D, то создаем образ v_i , он также будет выходом для F_D, и дугу α_i с ее меткой, направляем в этот выход. Примеры покрытий для автоматов, приведены на рисунках 2.а), .2.б) и 2.в).При проверке трансформационным методом двухленточных бинарных автоматов D_1 и D_2 на эквивалентность осуществляется следующее.

Строится покрытие автомата D_1, обозначим его $D_1(F)$. На покрытии отмечаются пары эквивалентных вершин. Все они заносятся в список пар эквивалентных вершин – $S = \{(s_1, s_1'), \mathbf{K}, (s_k, s_k')\}$.

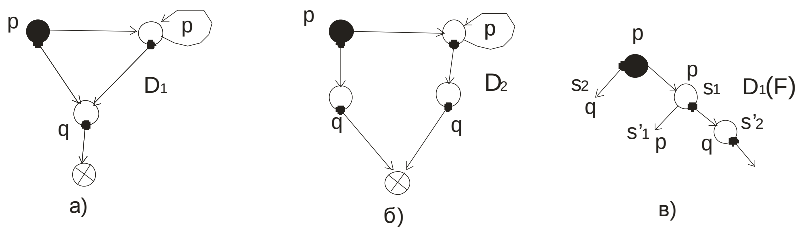

Рисунок 3 – Сравниваемые на эквивалентность автоматы D_1 и D_2 (а, б); покрытие автомата D_1 — $D_1(F)$ (в)

Затем алгоритм эквивалентными преобразованиями пытается преобразовать автомат D_2 в автомат D_3, который начинается древовидным фрагментом, назовем его куполом, изоморфным $D_1(F)$. На рисунке 3.а) и 3.б) приведены сравниваемые на эквивалентность автоматы D_1 и D_2. Покрытие автомата D_1 — $D_1(F)$ изображено на рисунке 3.в) и автомат D_2 преобразован в изображенный на рисунке 4 автомат D_3.

Если построение купола не удается, то алгоритм останавливается с заключением о том, что исходные автоматы D_1 и D_2 не эквивалентны. Пусть удалось в автомате D_2 построить купол, обозначаемый $D_2(F)$, изоморфный $D_1(F)$. Тогда на куполе отмечают пары вершин, которые являются образами эквивалентных вершин древовидного покрытия из списка S. Обозначим это множество пар вершин, принадлежащих куполу — $D = \{(d_1, d_1'), \mathbf{K}(d_k, d_k')\}$.

Далее алгоритм применяется к подмоделям, входы которых задаются парами вершин купола $(d_i, d_i'), i = 1, \mathbf{K}, n$. Этот процесс прослеживается на дереве потомков $T(D_1,D_2)$, корнем которого является пара сравниваемых моделей (D_1,D_2), а вершинами – пары сравниваемых подмоделей (рис. 4).

Сечение дерева $T(D_1,D_2)$ назовем *«золотым»*, если все его вершины помечены парами изоморфных автоматов. Если в дереве потомков имеется «золотое» сечение, то алгоритм классифицирует исходные модели как эквивалентные.

В работе [2] доказано, что трансформационный метод является алгоритмом разрешения проблемы эквивалентности для автоматов с непересекающимися циклами. При этом существенным является тот факт, что древовидное покрытие для автоматов с непересекающимися циклами определяется однозначно, что в общем случае это не так (см. примеры покрытий, приведенные на рис. 2.б) и 2.в)).

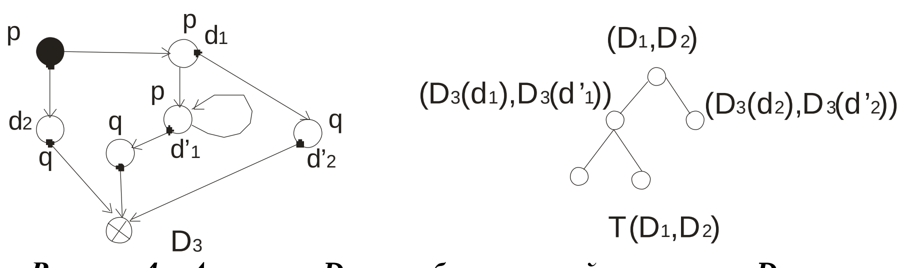

Рисунок 4 – Автомат D_3, преобразованный в автомат D_2

Автомат назовем *однозначным*, если он обладает единственным покрытием. Маршруты автомата, ведущие в одну вершину, назовем *параллельными*, если последние дуги этих маршрутов не совпадают и один из них не является началом другого. Вершину автомата назовем *правильной*, если в нее не ведут параллельные пути. Заметим, что покрытие строится неоднозначно, если в автомате имеются неправильные вершины, т.е вершины, в которые ведут простые маршруты, последние дуги которых различны.

Утверждение 1. Автомат является однозначным тогда и только тогда, когда все его вершины правильные.

Действительно, если a – неправильная вершина, тогда в нее приходят параллельные пути, а значит отличающиеся последними дугами. При построении покрытия автомата вершина a будет использована в одном ее покрытии с приходящей дугой, являющейся последней для одного из параллельных путей. Для второго покрытия эта же вершина a будет использована с приходящей дугой, являющейся последней для другого из параллельных путей.

В данной работе предложен алгоритм β, который по любому двухленточному бинарному автомату D_1 строит ему строго эквивалентный, но с однозначным покрытием.

Суть работы алгоритма заключается в избавлении от неправильных вершин, путем построения их копий. Причем считаем, что те дуги, которые выходят из исходных вершин, выходят и из их копий.

Схема алгоритма построения автомата с однозначным покрытием.

1. Помечается вход автомата.

2. Выбирается помеченная вершина, и просматриваются все выходящие из нее дуги:

   а) если дуга ведет в непомеченную вершину, то пометить дугу и инцидентную ей вершину;

   б) если дуга ведет в помеченную вершину, то:

   - если дуга не нарушит правильность вершины, тогда помечаем дугу и инцидентную ей вершину;
   - если дуга нарушает правильность вершины, тогда направляем ее в уже существующие копии вершины, если дуга не нарушит её правильность, иначе
   - создать копию вершины и дугу направить в нее. Из копии направить те же дуги, что и из исходной вершины.

3. Повторять шаг 2, до тех пор, пока есть помеченные вершины, выходящие дуги которых не помечены.

Рассмотрим пример, иллюстрирующий предложенный алгоритм.

На рисунке 5.а) показан исходный автомат, который не обладает однозначным покрытием, а на рисунке 5.б) строго эквивалентный исходному, но с однозначным покрытием.

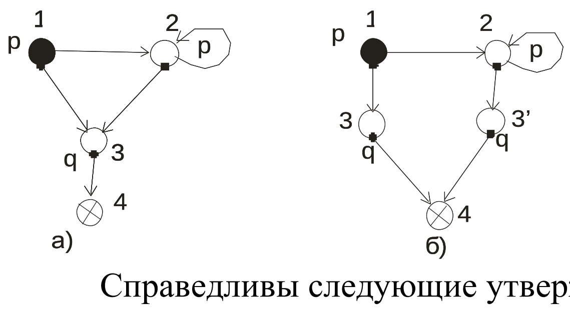

Рисунок 5 – Исходный автомат, который не обладает однозначным покрытием (а); автомат, строго эквивалентный исходному, но с однозначным покрытием (б)

Справедливы следующие утверждения.

Утверждение 2. На выходе алгоритма β будет построен автомат с однозначным покрытием.

Доказательство проведем от противного. Пусть D – результат работы алгоритма, и автомат D не является однозначным. Тогда в нем существует неправильная вершина *a*. Это означает, что в вершину *a* ведут два различных простых маршрута – l_1 и l_2. Пусть алгоритм вначале строит путь l_1, тогда при построении пути l_2, алгоритм должен построить дугу $(b,a) \in l_2$, причем вершина *a* – помеченная. Следовательно, дуга *(b,a)* нарушит однозначность вершины *a* и, согласно алгоритму, не может быть построена. Полученное противоречие и доказывает утверждение.

Утверждение 3. На выходе алгоритма β будет построен автомат с теми же простыми маршрутами, что и в исходном автомате.

Действительно, пусть алгоритм β по исходному автомату *D* построил автомат *D¢*. Рассмотрим в *D* простой путь l_1. Тогда для любой дуги *(a,b)* этого пути алгоритм β в автомате *D¢* построит дугу *(a¢ ,b¢)* , где *a¢* — это либо вершина *a*, либо ее копия, а *b¢* — это либо *b*, либо ее копия. Причем, согласно алгоритму, если на пути из входа уже построена некоторая вершина (или ее копия), то другие копии на этом пути больше не строятся, а следовательно простые маршруты сохраняются.

ЛИТЕРАТУРА

1. Подловченко Р.И., Хачатрян В.Е. Метод трансформационного распознавания эквивалентности в моделях вычислений // 8-ой межд.сем. «Дискретная математика и ее приложения». – МГУ – Москва, 2004. – С. 38-43.
2. Подловченко Р.И., Хачатрян В.Е. Об одном подходе к разрешению проблемы эквивалентности // Программирование. – 2004. – № 3. – С. 3-20.
3. Хачатрян В. Е. Трансформационный метод в моделях вычислений // Вестник компьютерных и информационных технологий. – 2008. – № 4. – С. 52-54.
4. Хачатрян В.Е., Великая Я.Г. Подход к решению фундаментальных проблем моделей вычислений на примере многоленточных автоматов // Научные ведомости БелГУ. Сер. История, Политология, экономика, Информатика. – 2009. – №9 (64). – С.108-111.
5. Хачатрян В.Е., Великая Я.Г. О трансформационном методе распознавания эквивалентности // Труды восьмой международной конференции «Дискретные модели в теории управляющих систем». – МГУ. – Москва, 2009. – С. 202-206.

Хачатрян Владимир Ервандович
Белгородский государственный университет, г. Белгород
Доктор физико-математических наук, доцент, профессор
Тел.: (4722) 30-13-53
E-mail: Khachatryan @bsu.edu.ru

Великая Яна Геннадьевна
Белгородский государственный университет, г. Белгород
Ассистент
Тел.: (4722) 30-13-53, 30-13-57
E-mail: velikaya@bsu.edu.ru

Сунцова Анастасия Игоревна
Белгородский государственный университет, г. Белгород
Ассистент
Тел.: (4722) 30-13-53, 30-13-57
E-mail:suntsova@bsu.edu.ru