



ОБОБЩЕННЫЙ ТРАНСФОРМАЦИОННЫЙ МЕТОД И КОНЕЧНЫЕ ДЕТЕРМИНИРОВАННЫЕ АВТОМАТЫ

Я.Г. ВЕЛИКАЯ

*Белгородский
государственный
университет*

e-mail: velikaya@bsu.edu.ru

Предложена модификация трансформационного метода, позволяющая решить проблему эквивалентности для конечных детерминированных автоматов.

Ключевые слова: трансформационный метод, проблема эквивалентности, конечный детерминированный автомат.

Модель вычислений в широком смысле может трактоваться как множество конструктивных объектов с приписанной ему универсальной процедурой, посредством которой каждому объекту сопоставляется порождаемое им множество.

Классическими моделями вычислений являются алгебраические выражения, формулы логики, конечные автоматы, абстрактные вычислительные машины, схемы программ, просто программы и другие.

Для моделей вычислений одной из фундаментальных проблем является проблема эквивалентности.

Эквивалентными в модели считаются объекты, порождающие одинаковые множества.

Отметим, что до настоящего времени не предложен алгоритм, решающий эту проблему для многоленточных автоматов, однако известно, что эта проблема разрешима [1].

Для решения проблемы эквивалентности многоленточных автоматов предложен трансформационный метод [2]. Было доказано, что данный метод решает проблему эквивалентности для некоторых подклассов многоленточных автоматов, в частности, для многоленточных автоматов с непересекающимися циклами [3].

С помощью трансформационного метода не удалось построить алгоритм, решающий проблему эквивалентности для конечных детерминированных автоматов [4].

В данной статье приводится модификация трансформационного метода и предлагается алгоритм, основанный на этой модификации, решающий проблему эквивалентности для конечных детерминированных автоматов.

Сформулируем основные понятия и опишем модификацию трансформационного метода. Предложим алгоритм и докажем, что он решает проблему эквивалентности для конечных автоматов.

Для простоты изложения материала будем рассматривать детерминированные конечные автоматы над алфавитом $Q=\{0,1\}$ и называть их *диаграммами*.

По определению, диаграмма – это конечный ориентированный граф с размеченными дугами. Его структура удовлетворяет следующим требованиям:

- в нем имеются две выделенные вершины, называемые входом и выходом диаграммы; из выхода нет исходящих дуг, а из всех остальных вершин исходит по две дуги;
- все дуги помечены символами алфавита Q , причем дуги, выходящие из одной вершины, помечены различными символами.

При изображении диаграмм условимся дуги, которые должны быть помечены символом 1, снабжать жирной точкой в начале дуги; метки дуг с символом 0 не отображать. Вход автомата будем обозначать черным кружком, а выход – перечеркнутым кружком. Пример диаграммы приведен на рис. 1.

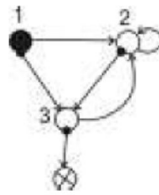


Рис. 1

Введем понятие эквивалентности диаграмм. Любой конечный ориентированный путь L в диаграмме может быть описан историей $L=(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$, где ϵ_i – это метка i -й дуги пути L . Путь, начинающийся и заканчивающийся в одной и той же вершине, назовем *циклом*. Определим *маршрут* [5] как путь, начинающийся во входе диаграммы. *Маршрутом через диаграмму* назовем маршрут, заканчивающийся в выходе диаграммы. Маршрут называется *простым*, если он не содержит циклов.

Диаграммы D_1 и D_2 назовем *эквивалентными*, если для любого маршрута L_1 через одну из диаграмм найдется маршрут L_2 через другую диаграмму, такие, что истории путей L_1 и L_2 совпадают.

Определим характеристику автомата D , называемую *покрытием* [6]. Это древовидный фрагмент, обозначим его $F(D)$, 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. 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.

Списком эквивалентных вершин покрытия $F(D)$ назовем список $S=\{(s_1, s'_1), \dots, (s_n, s'_n)\}$, где (s_i, s'_i) , $i=1, \dots, n$ – пара, состоящая из вершин, являющихся образом одной и той же вершины из списка V .

Отметим, что покрытие диаграммы в общем случае строится неоднозначно. На рис. 2 изображены покрытия диаграммы, приведенной на рис. 1.

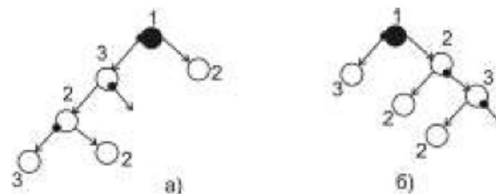


Рис. 2

Диаграмму назовем *однозначной*, если она обладает единственным покрытием. Опишем некоторые свойства однозначных диаграмм.

Маршруты диаграммы, ведущие в одну вершину, назовем *параллельными*, если последние дуги этих маршрутов не совпадают и один из маршрутов не является продолжением другого. Вершину диаграммы назовем *правильной*, если в неё не ведут параллельные маршруты. Вход и выход диаграммы условимся считать правильными.

В работе [7] описан алгоритм, назовем его алгоритмом γ , который преобразует любую диаграмму в эквивалентную ей однозначную диаграмму.

Приведем схему алгоритма γ построения однозначной диаграммы:

1) помечается вход диаграммы;

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 пометить дугу и инцидентную ей вершину,

• если дуга нарушает правильность вершины, тогда направить ее в уже существующие копии вершины, если дуга не нарушит её правильность, иначе создать копию вершины и дугу направить в нее. Из копии вывести те же дуги, что и из исходной вершины;

3) повторять пункт 2 до тех пор, пока есть помеченные вершины, выходящие дуги которых не помечены.

Однозначная диаграмма, построенная по диаграмме, изображенной на рис.1, приведена на рис. 3.

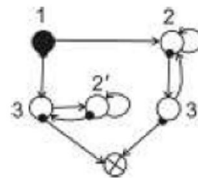


Рис. 3

Справедливы утверждения.

Утверждение 1 [7]. Если D – однозначная диаграмма и (s_i, s'_i) , $i=1, \dots, n$, – пара из списка S эквивалентных вершин покрытия $F(D)$, тогда вершина s_i на пути в $F(D)$ предшествует вершине s'_i .

Обозначим через D' диаграмму, полученную из диаграммы D по алгоритму γ .

Утверждение 2. Диаграммы D и D' таковы, что:

1) множества простых путей в диаграммах совпадают;

2) для любого простого цикла одной диаграммы в другой существует простой цикл с той же историей.

Доказательство пункта 1) имеется в работе [7]. Проведем доказательство пункта 2).

Рассмотрим в D некоторую вершину a , и пусть существует простой цикл L , образованный последовательностью дуг $(a,b),(b,c), \dots, (d,a)$. Тогда алгоритм γ для каждой дуги (a,b) цикла L построит в диаграмме D' дугу (a',b') , где a' – это образ вершины a ; b' – это образ вершины b . Причем, согласно алгоритму γ , если на пути из входа уже построен образ некоторой вершины или его копия, то другие копии вершины на этом пути больше не строятся, а следовательно, в D' будет построен простой цикл L' , образованный последовательностью дуг $(a',b'),(b',c'), \dots, (d',a')$. Понятно, что истории путей L и L' совпадают.

Докажем обратное. Пусть существует простой цикл L' , начинающийся в некоторой вершине a' в диаграмме D' и образованный последовательностью дуг $(a',b'),(b',c'), \dots, (d',a')$. Тогда в диаграмме D существует путь L , образованный последовательностью дуг $(a,b),(b,c), \dots, (d,a)$. Последнее объясняется тем, что дуга (a',b') могла быть



построена только в случае существования дуги (a,b) в диаграмме D , причем с той же меткой. Истории путей L и L' совпадают.

Опишем шаги процесса сравнения на эквивалентность диаграмм D_1 , D_2 , основанного на трансформационном методе [3].

Шаг 1. Построим покрытие $F(D_1)$ диаграммы D_1 и определим список $S = \{(s_1, s_1'), \dots, (s_n, s_n')\}$.

Шаг 2. Трансформируем диаграмму D_2 , используя эквивалентные преобразования [8], в диаграмму D_3 , начинающуюся куполом, изоморфным $F(D_1)$. Купол диаграммы – это дерево, состоящее из некоторых вершин и инцидентных им дуг диаграммы, корнем которого является вход диаграммы.

Если такое преобразование невозможно, то процесс заканчивает свою работу с заключением о том, что диаграммы D_1 и D_2 не являются эквивалентными. В противном случае строится список пар вершин $R = \{(r_1, r_1'), \dots, (r_n, r_n')\}$, каждый элемент которого является изоморфным образом вершин списка S .

Шаг 3. Выполним шаги 1-3 для каждой пары диаграмм, входы которых заданы вершинами списка R : $(D_3(r_1), D_3(r_1')), \dots, (D_3(r_n), D_3(r_n'))$.

Описанный процесс прослеживается на *дереве потомков* $T(D_1, D_2)$. Дерево потомков строится параллельно с вышеописанными шагами. Меткой корня дерева служит пара сравниваемых диаграмм (D_1, D_2) , а метками вершин – пары сравниваемых подграфов. Непосредственными потомками корня дерева $T(D_1, D_2)$ будут вершины, помеченные парами $(D_3(r_1), D_3(r_1')), \dots, (D_3(r_n), D_3(r_n'))$. У пар диаграмм $(D_3(r_i), D_3(r_i')), \dots, (D_3(r_n), D_3(r_n'))$ будут свои потомки и т.д.

Предложим следующую модификацию трансформационного метода:

- для каждой пары диаграмм дерева потомков в процессе сравнения диаграмм на эквивалентность необходимо выполнить дополнительный шаг, а именно первую из диаграмм предварительно преобразовать в однозначную диаграмму.

Обозначим через μ алгоритм сравнения на эквивалентность двух диаграмм, использующий модификацию трансформационного метода. Такой алгоритм легко получить из описанного процесса, алгоритмизируя неоднозначность выбора сравниваемых на эквивалентность диаграмм дерева потомков.

Теорема. Алгоритм μ является алгоритмом разрешения проблемы эквивалентности диаграмм.

Легко видеть, что доказательство теоремы следует из справедливости следующих двух лемм.

Лемма 1. Если диаграммы D_1 и D_2 – эквивалентны, то в дереве потомков $T(D_1, D_2)$ непременно имеется α -сечение.

Сечение дерева $T(D_1, D_2)$ называется α -сечением, если все вершины сечения помечены парами изоморфных диаграмм.

Лемма 2. α -сечение в дереве потомков $T(D_1, D_2)$ строится за конечное число шагов, которое задается только количеством вершин в диаграммах D_1 и D_2 .

Доказательство леммы 1. Пусть алгоритм μ , примененный к паре (D_1, D_2) , выявил пары поддиаграмм, требующие проверки на эквивалентность $(D_3(r_i), D_3(r_i'))$, $i=1, \dots, n$, где D_3 – диаграмма, эквивалентная D_2 и начинающаяся куполом, изоморфным $F(D_1')$, где $F(D_1')$ – покрытие диаграммы D_1' , причем D_1' – однозначная диаграмма, эквивалентная D_1 .

Из эквивалентности диаграмм D_1 и D_2 следует, что любые пары $(D_3(r_i), D_3(r_i'))$, $i=1, \dots, n$ состоят из эквивалентных диаграмм. Согласно утверждению 1, вершина r_i предшествует вершине r_i' . Т.к. r_i и r_i' являются входами эквивалентных диаграмм, то существует вершина r_i'' , которой предшествует вершина r_i' такая, что диаграмма $D_3(r_i')$ эквивалентна диаграмме $D_3(r_i'')$. История пути между вершинами r_i и r_i' , обозначим его L' , совпадает с историей пути между вершинами r_i' и r_i'' . Поскольку число вершин диаграммы D_3 , входы которых задают эквивалентные диаграммы, конечно, то существует вершина



r_i^* , такая что $D_3(r_i)$ и $D_3(r_i^*)$ эквивалентны, причем r_i^* является началом и концом цикла, обозначим его L . Истории путей L и L' таковы, что одна из них непременно кратна другой. Это означает, что при проверке на эквивалентность пары диаграмм $(D_3(r_i), D_3(r_i'))$ алгоритм μ обязательно найдет потомка $(D_3(r_i^*), D_3(r_i^{**}))$, метка которого содержит пару изоморфных поддиаграмм. Аналогичные рассуждения можно проделать для любой пары поддиаграмм, являющихся меткой любой вершины дерева потомков $T(D_1, D_2)$. Количество таких пар конечно, поскольку они определяют циклы в диаграмме. Следовательно, при проверке потомка пары (D_1, D_2) алгоритм μ найдет на каждой ветви дерева потомков пару изоморфных поддиаграмм, т.е. построит в дереве потомков $T(D_1, D_2)$ α -сечение.

Для доказательства леммы 2 предварительно докажем утверждение 3.

Пусть D_1 и D_2 – пара сравниваемых на эквивалентность диаграмм. Диаграмма D_1 приводится к однозначной диаграмме D_1' с покрытием $F(D_1')$ и списком $S = \{(s_1, s_1'), \dots, (s_n, s_n')\}$. Обозначим через V множество поддиаграмм диаграммы D_2 . Диаграмма D_2 преобразуется в D_3 с куполом, изоморфным $F(D_1')$. Вершины купола, изоморфные вершинам из списка S , обозначим списком $R = \{(r_1, r_1'), \dots, (r_n, r_n')\}$.

Утверждение 3. Если (D_1, D_2) – пара диаграмм, сравниваемых на эквивалентность, с количеством вершин n_1 и n_2 соответственно, тогда любой потомок (D'', D') вершины, помеченной парой $(D_3(r_i), D_3(r_i'))$, $i=1, \dots, n$, удовлетворяет условиям:

- D'' начинается куполом, на листьях которого расположены диаграммы из множества V , а высота купола не превышает n_2 ;
- D' принадлежит множеству V .

Доказательство. При сравнении пары $D_3(r_i), D_3(r_i')$ на эквивалентность диаграмма $D_3(r_i)$ будет преобразована к однозначной. Обозначим её $D_3'(r_i)$. Согласно утверждению 2 диаграмма $D_3'(r_i)$ содержит циклы с той же историей, что и $D_3(r_i)$. Легко видеть, что в диаграмме D_3 содержатся циклы с той же историей, что и в D_2 . Поскольку длина пути, являющегося циклом диаграммы D_2 , ограничена числом n_2 , то и длина циклов, принадлежащих диаграмме D_3 , а значит и поддиаграмме $D_3(r_i)$, ограничена этим же числом. Следовательно, длина циклов диаграммы $D_3'(r_i)$ ограничена числом n_2 . Из утверждения 1 следует, что для покрытия $F(D_3'(r_i))$ диаграммы $D_3'(r_i)$ в списке $M = \{(m_1, m_1'), \dots, (m_t, m_t')\}$ пар эквивалентных вершин, первая вершина пары предшествует второй, причем путь между эквивалентными вершинами m_i, m_i' , где $i=1, \dots, t$, не превышает n_2 . Диаграмма $D_3(r_i')$, принадлежащая множеству V , будет преобразована в диаграмму $D_4(r_i')$, начинающуюся куполом, изоморфным покрытию $F(D_3'(r_i))$, с некоторым списком пар эквивалентных вершин $K = \{(k_1, k_1'), \dots, (k_t, k_t')\}$. Диаграмма $D_4(k_i')$, $i=1, \dots, t$ принадлежит множеству V , и для любой пары из списка K длина пути между вершинами (k_i, k_i') , $i=1, \dots, t$ ограничена числом n_2 . Аналогичное рассуждение можно провести для потомков пары $(D_4(k_i), D_4(k_i'))$, где $i=1, \dots, t$ и всех последующих.

Доказательство леммы 2. Согласно лемме 1, в дереве потомков $T(D_1, D_2)$ будет построено α -сечение. Из утверждения 3 следует, что число потомков вершины, помеченной парой $(D_3(r_i), D_3(r_i'))$, $i=1, \dots, n$ – конечно и зависит от количества вершин диаграммы D_2 . Это означает, что α -сечение в дереве потомков $T(D_1, D_2)$ строится конечным числом шагов.

Рассмотрим пример, приведенный в работе [4], для которого применение трансформационного метода не позволило в дереве потомков получить α -сечение. Покажем, что предложенный нами алгоритм, основанный на обобщенном трансформационном методе, позволяет для этой пары диаграмм получить в дереве потомков α -сечение. Пара диаграмм, взятая из [4], приведена на рис. 4.

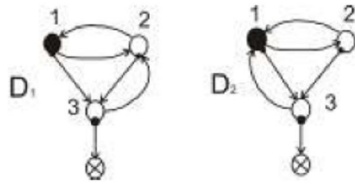


Рис. 4

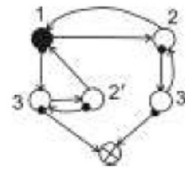


Рис. 5

Для диаграммы D_1 построим эквивалентную ей диаграмму D_1' с однозначным покрытием (рис. 5).

Покрытие $F(D_1')$ со списком пар эквивалентных вершин $A = \{(a_1, a_1'), (a_1, a_1''), (a_2, a_2'), (a_3, a_3')\}$ – изображено на рис. 6.

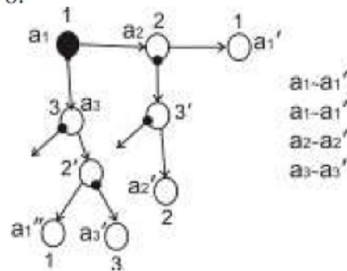


Рис. 6

Эквивалентными преобразованиями алгоритм пытается преобразовать диаграмму D_2 в диаграмму D_3 , которая начинается куполом, изоморфным $F(D_1')$.

Если построение купола не удастся, то алгоритм останавливается с заключением о том, что исходные диаграммы D_1 и D_2 , не эквивалентны. В данном примере алгоритм смог построить диаграмму D_3 , начинающуюся куполом, изоморфным $D_1'(F)$ (рис. 7). На рис. 8 приведен купол диаграммы D_3 , изоморфный $D_1'(F)$. На куполе диаграммы D_3 отмечены пары вершин, которые являются образами эквивалентных вершин древовидного покрытия из списка A , принадлежащие списку B : $B = \{(b_1, b_1'), (b_1, b_1''), (b_2, b_2'), (b_3, b_3')\}$.

В списке B хранятся пары вершин, «претендующих» на эквивалентность. Причем видно (рис. 8), что пары вершин (b_1, b_1') и (b_3, b_3') содержат эквивалентные вершины, т.к. этими вершинами обозначено одно и то же состояние диаграммы D_3 .

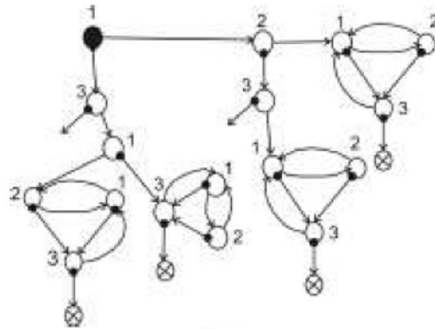


Рис. 7

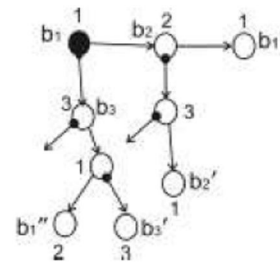


Рис. 8

Далее алгоритм применяется к поддиаграммам, входы которых задаются парами вершин из списка B , эквивалентность которых необходимо доказать: (b_1, b_1') и (b_2, b_2') .

Рассмотрим применение алгоритма к первой паре поддиаграмм.

Применим алгоритм к поддиаграммам $((D_3(b_1), D_3(b_1''))$ (рис. 9).

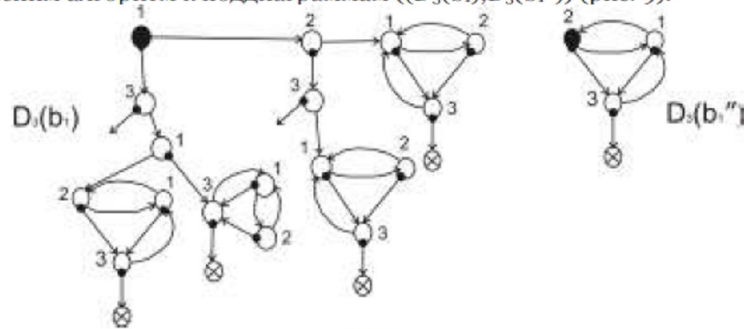


Рис. 9

Алгоритм приводит первую из диаграмм, диаграмму $D_3(b_1)$, к эквивалентной ей диаграмме, обладающей однозначным покрытием, обозначим ее $D_3'(b_1)$ (рис. 10).

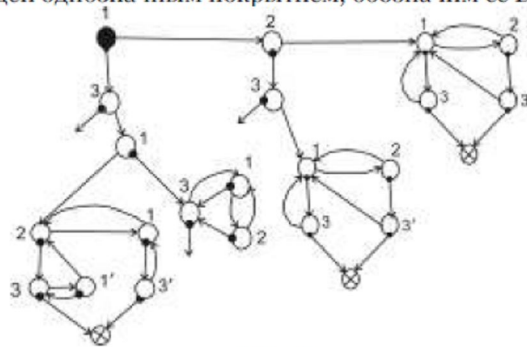


Рис. 10

Далее алгоритм построит покрытие диаграммы $D_3'(b_1)$, причем для поддиаграммы с входом в вершине b_1' покрытие не будет построено, поскольку вершины, изоморфные паре (b_1, b_1') , должны быть эквивалентны. Так же не будет построено покрытие для поддиаграммы с входом в вершине b_3' , поскольку вершины, изоморфные паре (b_3, b_3') , должны быть эквивалентны.

На рис. 11 изображено древовидное покрытие диаграммы $D_3'(b_1)$ и построен список эквивалентных вершин $C = \{(c_1, c_1'), (c_2, c_2'), (c_2, c_2''), (c_3, c_3'), (c_4, c_4'), (c_5, c_5'), (c_6, c_6'), (c_6, c_6''), (c_6, c_6''')\}$.

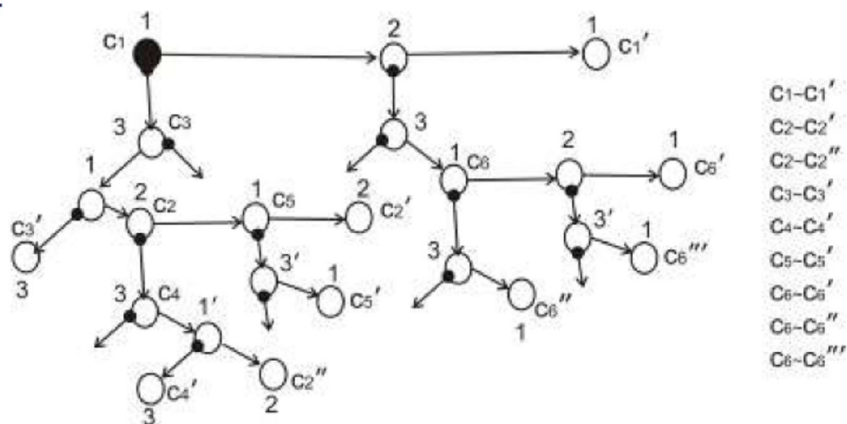


Рис. 11



Далее алгоритм пытается преобразовать диаграмму $D_3(b_1'')$ в D_4 с куполом, изоморфным $F(D_3'(b_1))$. Если построение купола не удастся, то алгоритм останавливается с заключением о том, что исходные диаграммы D_1 и D_2 не эквивалентны. В данном случае алгоритм построил вышеописанную диаграмму D_4 . Купол диаграммы D_4 приведен на рис. 12. Списком пар вершин, являющихся изоморфными образами вершин из списка C , будет список $M = \{(m_1, m_1'), (m_2, m_2'), (m_2, m_2''), (m_3, m_3'), (m_4, m_4'), (m_5, m_5'), (m_6, m_6'), (m_6, m_6''), (m_6, m_6''')\}$.

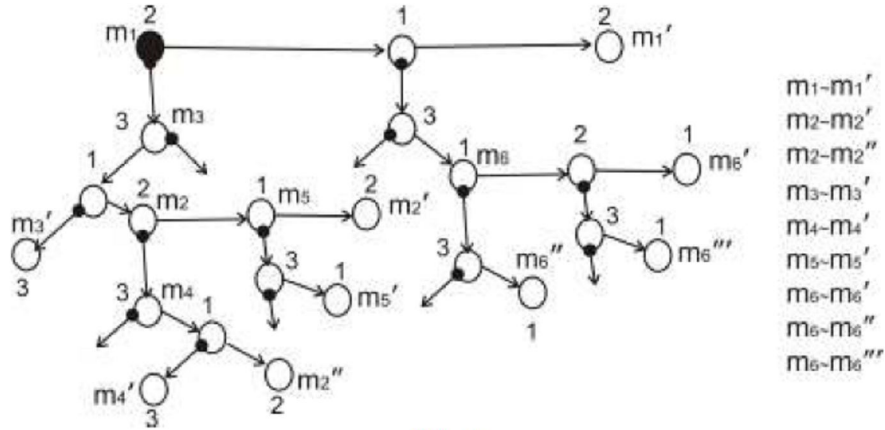


Рис. 12

Так как пары вершин из списка M являются входами эквивалентных диаграмм, то на следующем шаге алгоритм будет применен к каждой паре поддиаграмм, и алгоритм завершится с заключением об эквивалентности диаграмм $D_3(b_1)$ и $D_3(b_1'')$.

Затем алгоритм применяется к поддиаграммам, входы которых задаются второй парой вершин из списка B , «претендующих» на эквивалентность: (b_2, b_2') . Т.е. алгоритм применится к паре поддиаграмм $((D_3(b_2), D_3(b_2'))$ (рис. 13).

Для диаграммы $D_3(b_2)$ алгоритм построит эквивалентную однозначную диаграмму, обозначим её $D_3'(b_2)$. Диаграмма $D_3'(b_2)$ изображена на рис. 14.

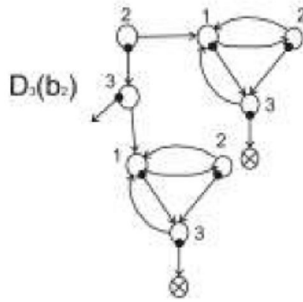


Рис. 13

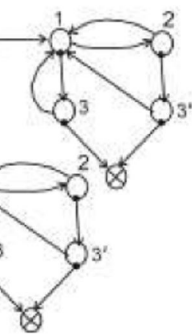
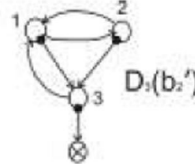


Рис. 14

А после этого алгоритм построит покрытие диаграммы $D_3'(b_2)$, обозначим его $F(D_3'(b_2))$, и список пар эквивалентных вершин – $K = \{(k_1, k_1'), (k_1, k_1''), (k_1, k_1'''), (k_2, k_2'), (k_2, k_2''), (k_2, k_2''')\}$ (рис. 15).

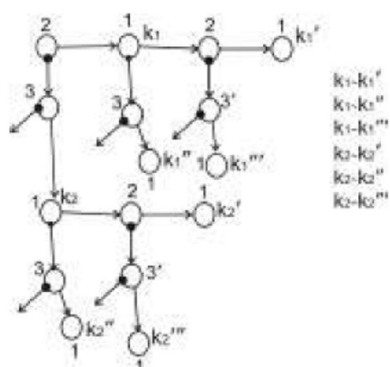


Рис. 15

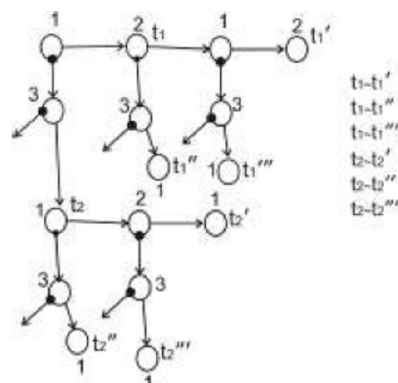


Рис. 16

Далее алгоритм пытается преобразовать диаграмму $D_3(b_2')$ в диаграмму D_5 с куполом, изоморфным $F(D_3'(b_2))$. Купол диаграммы D_5 приведен на рис. 16. Списком пар вершин, являющихся изоморфными образами вершин из списка K , будет список $T = \{(t_1, t_1'), (t_1, t_1''), (t_1, t_1'''), (t_2, t_2'), (t_2, t_2''), (t_2, t_2''')\}$.

Далее алгоритм применяется к поддиаграммам, входы которых задаются парами вершин из списка T , «претендующих» на эквивалентность, эквивалентность которых необходимо доказать. Это пары поддиаграмм: $(D_5(t_1), D_5(t_1'''))$ и $(D_5(t_1), D_5(t_1''))$.

Рассмотрим процесс применения алгоритма к паре $(D_5(t_1), D_5(t_1'''))$ (рис. 17).

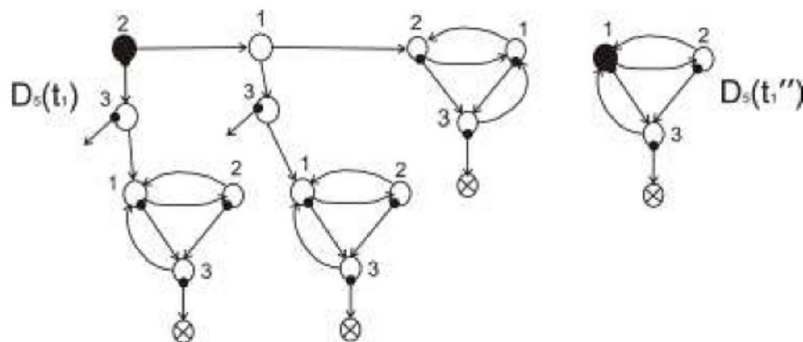


Рис. 17

Для диаграммы $D_5(t_1)$ будет построена однозначная диаграмма, которая изображена на рис. 18, обозначим её $D_5'(t_1)$.

Далее алгоритм построит покрытие диаграммы $D_5'(t_1)$, причем для поддиаграммы с входом в вершине t_1' покрытие не будет построено, поскольку вершины, изоморфные паре (t_1, t_1') , должны быть эквивалентны.

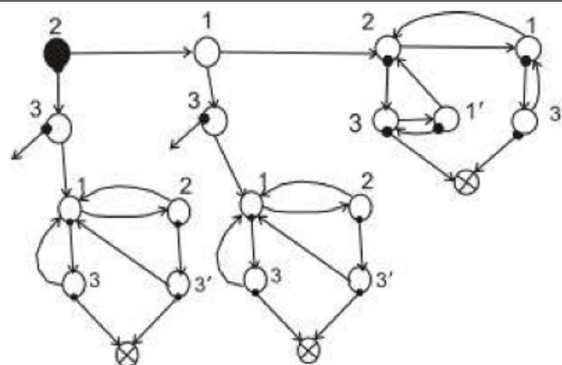


Рис. 18

На рис. 19 изображено древовидное покрытие диаграммы $D_5(t_1)$ и построен список эквивалентных вершин $N = \{(n_1, n_1'), (n_2, n_2'), (n_2, n_2''), (n_2, n_2'''), (n_3, n_3'), (n_3, n_3''), (n_3, n_3''')\}$.

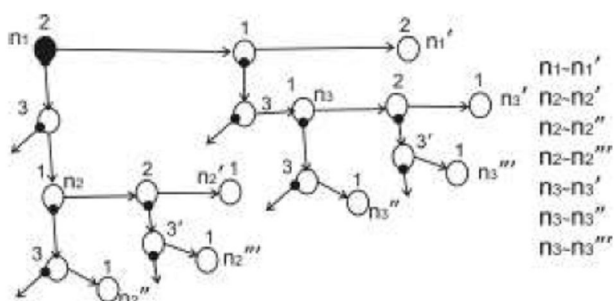


Рис. 19

Далее алгоритм пытается преобразовать диаграмму $D_5(t_1)$ в D_6 с куполом, изоморфным покрытию диаграммы $D_5(t_1)$. Купол диаграммы D_6 приведен на рис. 20. Списком пар вершин, являющихся изоморфными образами вершин из списка N , будет список $L = \{(l_1, l_1'), (l_2, l_2'), (l_2, l_2''), (l_2, l_2'''), (l_3, l_3'), (l_3, l_3''), (l_3, l_3''')\}$.

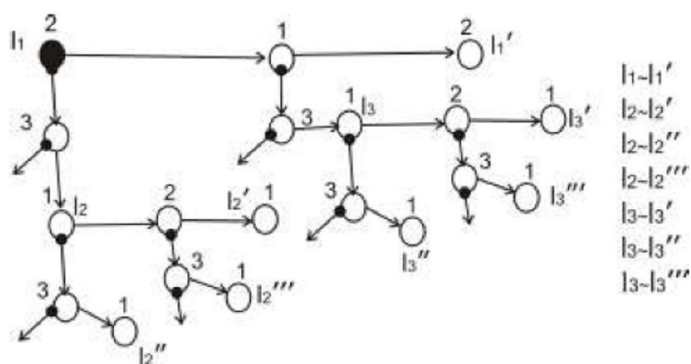


Рис. 20

Поскольку пары вершин из списка L являются входами эквивалентных диаграмм, то на следующем шаге алгоритм будет применен к каждой паре поддиаграмм, и алгоритм завершится с заключением об эквивалентности диаграмм $D_5(t_1)$ и $D_5(t_1')$.



Применение алгоритма к паре $(D_5(t_1), D_5(t_1''))$ аналогично его применению к паре $(D_5(t_1), D_5(t_1'))$ в виду того, что пары поддиаграмм $(D_5(t_1), D_5(t_1'))$ и $(D_5(t_1), D_5(t_1''))$ образованы одними и теми же диаграммами.

Алгоритм завершится с заключением об эквивалентности исходных диаграмм D_1 и D_2 .

Литература

1. Harju T., Karhumaki J. The equivalence of multitape finite automata // Theoret. Computer Sci. 1991. 78. P. 347-355.
2. Подловченко Р.И., Хачатрян В.Е. Метод трансформационного распознавания эквивалентности в моделях вычислений // 8-й межд. сем. «Дискретная математика и ее приложения». – М.: МГУ, 2004. С. 38-43.
3. Подловченко Р.И., Хачатрян В.Е. Об одном подходе к разрешению проблемы эквивалентности // Программирование. 2004. № 3. С. 3-20.
4. Хачатрян В. Е. Трансформационный метод в моделях вычислений // Вестник компьютерных и информационных технологий. 2008. № 4. С. 52-55.
5. Подловченко Р.И., Айрапетян М.Г. О построении полных систем эквивалентных преобразований схем программ // Программирование. 1996. № 1. С. 3-29.
6. Хачатрян В.Е., Великая Я.Г. Модели вычислений с однозначным покрытием // Научные ведомости БелГУ. 2009. № 7(62). С.116-121.
7. Хачатрян В.Е., Великая Я.Г., Сунцова А.И. О преобразовании модели вычислений к однозначному виду // Информационные системы и технологии. 2010. № 3(59). С.45-49.
8. Хачатрян В. Е. Полная система эквивалентных преобразований для многоленточных автоматов // Программирование. 2003. №1. С. 62-77.

THE GENERALIZED TRANSFORMATIONAL METHOD AND FINITE DETERMINISTIC AUTOMATA

Y.G. VELIKAYA

Belgorod State University

e-mail: velikaya@bsu.edu.ru

The transformational method updating is offered, allowing to solve a problem of equivalence for finite deterministic automata.

Key words: the transformational method, a problem of equivalence, finite deterministic automata.