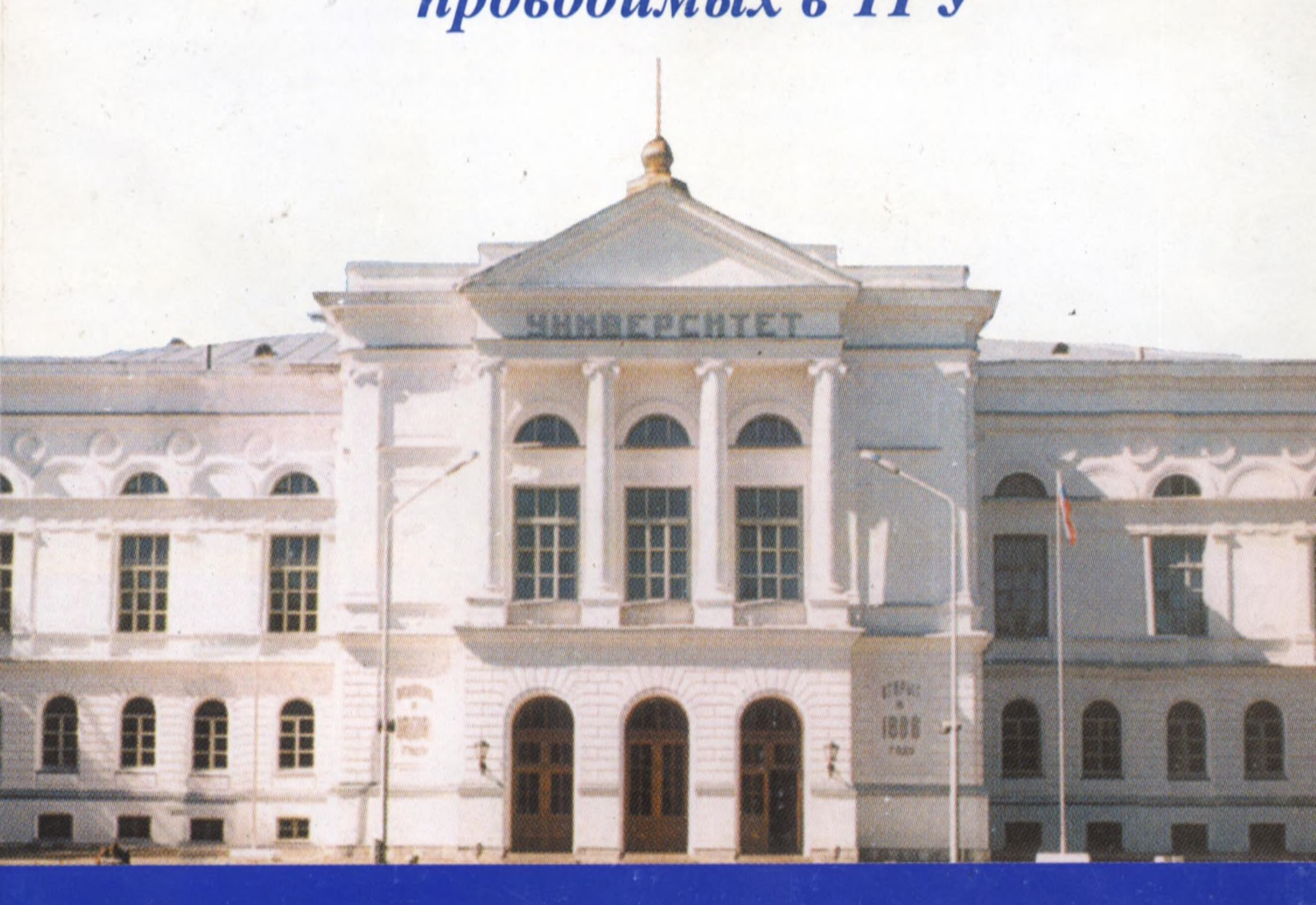
ISSN 1561-7793

Вестник
Томского государственного университета

ПРИЛОЖЕНИЕ

№ 18 АВГУСТ 2006

Материалы международных, всероссийских и региональных научных конференций, симпозиумов, школ, проводимых в ТГУ



Уникальные идентификаторы состояний протокола суть $UIO_1 = \{CR\}$, $UIO_2 = \{ICONrsp\}$, $UIO_3 = UIO_4 = \{DT0\}$ или $UIO_3=UIO_4=\{DT1\}$. Длина каждого уникального идентификатора не превышает 1.

Заключение

В работе изложен подход к пассивному тестированию, основанный на сокращении спецификации, в том числе с использованием уникальных идентификаторов состояний. Показано, что автомат-спецификация может быть сокращен, во-первых, за счет построения *l*-эквивалента автомата. Поскольку такой автомат является недетерминированным, то можно однозначно установить начальное состояние спецификации, используя уникальные идентификаторы состояний, наблюдая их во входо-выходной последовательности и проверяя, какому состоянию спецификации соответствует данный идентификатор.

ЛИТЕРАТУРА

1. *Гилл А.* Введение в теорию конечных автоматов. – М.: Наука, 1966.– 272 с.
2. *Трахтенброт Б.А., Барздинь Я.М.* Конечные автоматы (Поведение и синтез). – М.: Наука, 1970. – 400 с.
3. *Sabnani K., Dahbura A.* A protocol test generation procedure // Computer Networks and ISDN Systems. – 1988. – V. 15. – No. 4. – P. 283 – 297.
4. *Dorofeeva M., El-Fakih K., Cavalli A.R., Yevtushenko N.* Experimental evaluation of FSM-based testing methods // Proceedings IEEE International Conference on Software Engineering and Formal Methods. – 2005. – P. 23 – 32.
5. *Hogrefe D.* OSI Formal Specification Case Study: The Inres Protocol and Service // Technical report IAM-91-012: University of Bern. – 1991.

УДК 618.5:519.68

МЕТОД СИНТЕЗА ПРОВЕРЯЮЩИХ ТЕСТОВ ДЛЯ РАСШИРЕННЫХ АВТОМАТОВ БЕЗ ПОСТРОЕНИЯ ЭКВИВАЛЕНТНОГО КОНЕЧНОГО АВТОМАТА

А.В. Коломеец, С.А. Прокопенко

Томский государственный университет

E-mail: {786-kav, prokopenko}@elefot.tsu.ru

Предлагается метод синтеза проверяющих тестов для расширенных автоматов на основе частичного моделирования поведения автомата.

Ключевые слова: *расширенный автомат, конечный автомат, проверяющий тест.*

Введение

Построение проверяющих тестов для протоколов вычислительных сетей является одной из актуальных технических задач. Для построения проверяющего теста необходимо иметь, в первую очередь, адекватную математическую модель протокола. В настоящее время для описания поведения протоколов вычислительных сетей все чаще используются расширенные автоматы [1 – 3].

В большинстве случаев при построении проверяющих тестов поведение расширенного автомата моделируется на параметризированных входных последовательностях, и далее тест строится известными методами по эквивалентному конечному автомату. Однако такой переход не всегда возможен, а в тех случаях, когда такой переход возможен, размерность эквивалентного конечного автомата получается очень большой. В данной работе мы показываем, что в случае, когда состояния расширенного автомата обладают уникальными идентификаторами, при построении теста с фиксированной полнотой можно ограничиться частичным моделированием расширенного автомата.

1. Определения

Формально [1] под *расширенным автоматом M* понимается пятерка (S,X,Y,T,V), где S – непустое конечное множество состояний автомата; X – непустое множество входных символов, называемое входным алфавитом; Y – непустое множество выходных символов, называемое выходным алфавитом; V – конечное, возможно, пустое множество контекстных переменных; T – множество переходов между состояниями из S.

Каждый переход t из T – это семёрка (s,x,P,op,y,up,s'), где s и s' принадлежат множеству состояний S и являются начальным и конечным состояниями перехода; $x \in X$ есть входной символ; $y \in Y$ – выходной символ; P, op и up – функции, определенные над входными параметрами и контекстными переменными из V:

P: $D_{inp}x \times D_V \rightarrow$ {Истина, Ложь} – предикат, где D_V – множество контекстных векторов;

op: $D_{inp}x \times D_V \to D_{out}y$ – функции расчета выходного параметра;

up: $D_{inp}x \times D_V \to D_V$ – функция изменения значения контекстной переменной.

Здесь $D_{inp}x$ обозначает множество входных векторов, компонентами которых являются значения параметров, соответствующих входному символу x (далее входные параметры); $D_{out}y$ обозначает множество выходных векторов, компонентами которых являются значения параметров, соответствующих выходному символу y (далее выходные параметры);

Параметризованным входным символом называется пара «входной символ x, входной вектор **a** из $D_{inp}x$», т.е. пара (x,a).

Конфигурацией расширенного автомата M называется пара «состояние s, контекстный вектор v», т.е. (s,v).

Переход называется *возбужденным* для конфигурации (s,v) и параметризованного входа (x,a), если предикат перехода принимает значение «Истина» для пары (v,a).

Расширенный автомат, как и конечный, графически может быть представлен диаграммой переходов. На рис. 1 приведена диаграмма переходов расширенного автомата SC, описывающего поведение протокола Simple Connection [2].

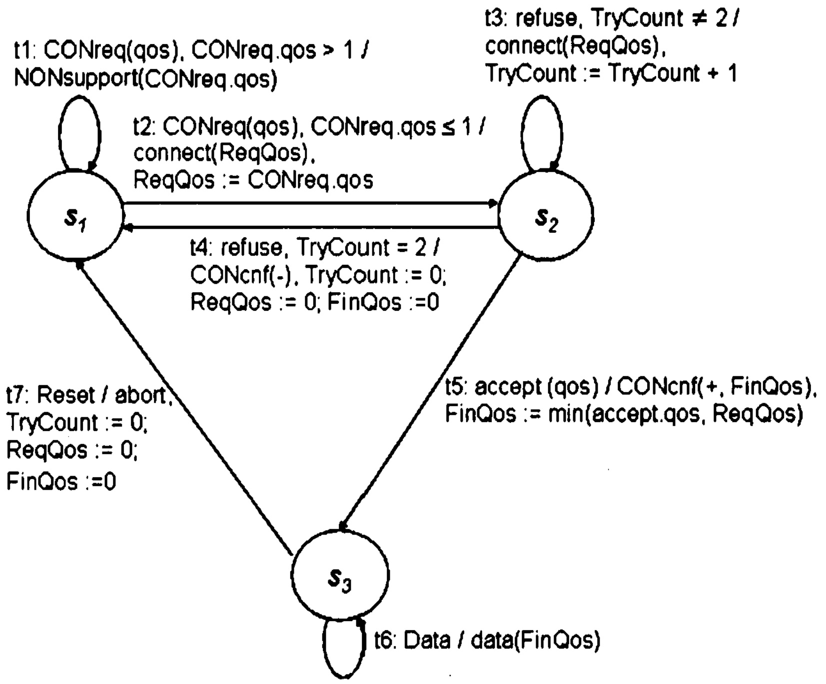


Рис. 1. Диаграмма переходов расширенного автомата, описывающего поведение протокола Simple Connection

Рассмотрим работу расширенного автомата на примере. Пусть автомат находится в начальном состоянии s_1 и на вход автомата поступает входной символ CONreq с входным параметром qos. Из состояния s_1 под действием входного символа CONreq возможны два перехода: один в состояние s_1 и второй – в состояние s_2. Поскольку контекстная переменная CONreq.qos равна 0 и значение контекстной переменной удовлетворяет значению предиката CONreq.qos≤1 второго перехода, то из состояния s_1 возможен лишь переход в состояние s_2. В этом случае автомат обновляет значение другой контекстной переменной ReqQos в соответствии с назначением ReqQos=CONreq.qos, производит выходной символ connect с выходным параметром ReqQos и переходит в состояние s_2.

Автомат A называется *полностью определенным*, если для каждой пары $(x,s)\in X\times S$ существует, по крайней мере, одна пара $(y,s')\in Y\times S$ такая, что $(s,x,P,op,y,up,s')\in T$, где P,op,up – любые. В противном случае автомат A называется *частичным*.

Если автомат является не полностью определенным, то неопределенные переходы обычно доопределяются как петля в соответствующем состоянии со специальным выходным символом «null» или «ignore».

Известно [1], что в случае, когда области определения контекстных переменных и входных параметров расширенного автомата конечны, то можно построить конечный автомат, эквивалентный расширенному автомату, путем моделирования поведения последнего. Это значит, что на любую входную последовательность выходные последовательности обоих автоматов будут совпадать. Далее рассмотрим расширенные автоматы, для которых можно построить эквивалентный конечный автомат, и опишем модель неисправности для построения диагностических тестов.

2. Построение по расширенному автомату эквивалентного конечного автомата

В расширенном автомате каждый переход определяется не только текущим состоянием автомата и параметризованным входным символом, но и значениями контекстных переменных в данном состоянии, поскольку каждый переход зависит от того, удовлетворяется ли предикат данного перехода. Иными словами, каждое состояние расширенного автомата определяется конфигурацией, т.е. парой «состояние, набор значений контекстных переменных». Контекстные переменные определяются, в свою очередь, функциями изменения значений контекстной переменной и параметрами входных символов. Таким образом, каждый переход $t = (s, x, P, op, y, up, s')$ расширенного автомата может быть представлен множеством переходов $\{((s,v),x',y',(s',v'))\}$. Здесь x' обозначает пару (входной символ, значение входного параметра), y' обозначает пару (выходной символ, значение выходного параметра). Каждый переход вида $((s,v),x',y',(s',v'))$ можно интерпретировать как переход конечного автомата, так как его входы и выходы не зависят от параметров, а каждое состояние является конфигурацией расширенного автомата. Таким образом, по расширенному автомату можно построить эквивалентный ему конечный автомат, в котором множество состояний есть множество конфигураций расширенного автомата, которые достижимы из начальной конфигурации, множество входных символов есть множество пар (входной символ, значение входного параметра) и множество выходных символов – множество пар (выходной символ, значение выходного параметра).

Конечный автомат A_M можно построить по расширенному автомату $M=(S,T)$ и более простым способом, оставив на каждом переходе автомата только входной и выходной символы (без параметров, предикатов и функций изменения контекстной переменной и выходных параметров). В общем случае такой автомат получается недетерминированным и частично определенным. Однако если исходный расширенный автомат полностью определенный, то и соответствующий конечный автомат будет полностью определенным.

Построим для автомата A_M семейство гармонизированных идентификаторов. Множество различающих последовательностей $w(s_i,s_j)$ по всем состояниям s_j автомата A_M, которые различимы с состоянием s_i, называется *идентификатором* состояния s_i и обозначается $d(s_i)$. Семейство множеств $\{d(s_i): s_i \in S\}$ называется *семейством идентификаторов* автомата A. Семейство $H = \{H_1,\ldots,H_n\}$ идентификаторов состояний эталонного автомата A_M называется *семейством гармонизированных идентификаторов*, если для каждой пары состояний s_i, $s_j \in S$ в пересечении идентификаторов H_i и H_j существует последовательность, которая различает состояния s_i и s_j. Для каждого приведенного полностью определенного или частичного конечного автомата можно построить семейство гармонизированных идентификаторов.

Семейство гармонизированных идентификаторов для конечного автомата, построенного по автомату (рис. 1), изображенному на рис. 2, состоит из следующих входо-выходных пар: CONreq/NONsupport, CONreq/connect, accept/CONcnf2, Data/data.

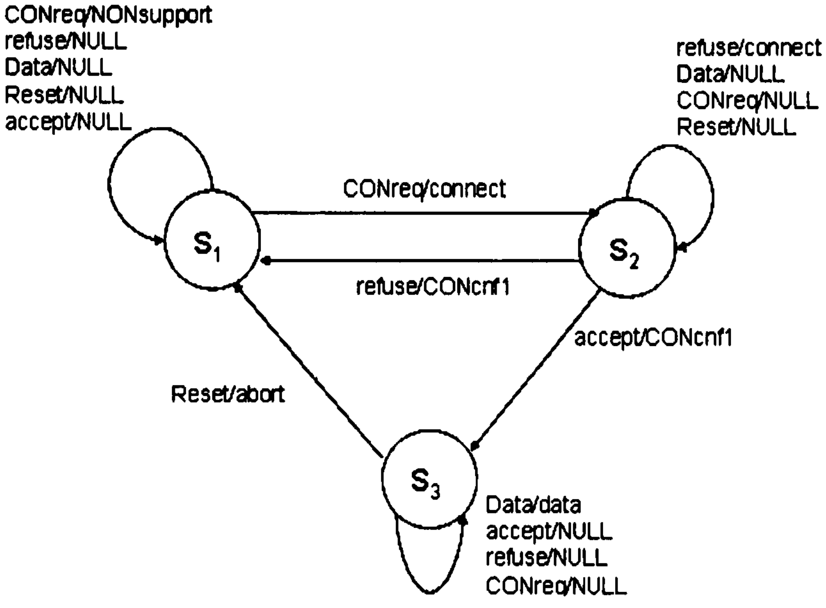


Рис. 2. Диаграмма переходов SCP с переходами, доопределенными нулевым выходом

3. Модель неисправности для расширенного автомата

Построение тестов для дискретных систем, поведение которых описано автоматами, обычно требует анализа автомата как в исправном, так и в неисправном состоянии. Прежде всего, необходимо договориться, какого рода неисправности будем рассматривать. Иными словами, определим модель неисправности.

Под *неисправностью* в автомате будем понимать неисправности на выходе или на переходах, т.е. преобразование эталонного автомата в автомат с другими функциями переходов или выходов.

Неисправность на выходе [4] возникает, когда автомат из текущего состояния под воздействием входного символа переходит в ожидаемое состояние, но с неожидаемым выходным символом. *Неисправность на переходе* [4] возникает, когда автомат из текущего состояния под воздействием входного символа переходит в неожидаемое состояние.

Пусть $M=(S,T)$ – расширенный автомат, описывающий поведение спецификации, и $\mathfrak{I}=\{I=(S,P)\}$ – множество автоматов-реализаций. Мы полагаем, что число состояний в проверяемом расширенном автомате равно числу состояний автомата-спецификации и автомат-реализация – полностью определенный. Выделим в автомате-спецификации все переходы, которые не имеют условий, то есть переходы без предикатов, и построим соответствующий конечный автомат A_M на этих переходах. Автомат строится простым удалением переходов с предикатами и удалением функции вычисления нового значения контекстной переменной на остальных переходах. Далее полагаем, что полученный автомат A_M является приведенным и связным.

4. Синтез проверяющих тестов для расширенных автоматов относительно выходных неисправностей без полного моделирования расширенного автомата

В данном разделе мы предлагаем метод синтеза тестов для класса расширенных автоматов с гарантированной полнотой. Метод не требует полного построения эквивалентного конечного автомата.

Теорема. Пусть $H=\{H,\ldots,H_n\}$ – семейство гармонизированных идентификаторов автомата A_M; TS – множество параметризованных входных последовательностей расширенного автомата-спецификации; $A(M)$ – конечный автомат, эквивалентный спецификации M. Если для любого перехода с начальным состоянием s_i, финальным s_j и входным символом x автомата M в множестве TS существуют подмножества $\alpha_i H_i$ и $\alpha_i x H_j$, то TS обнаруживает любую реализацию из множества $\mathfrak{I}$, которая неэквивалентна M.

Теорема дает достаточные условия полноты теста, который построен на основе прохождения переходов автомата M. Заметим, что идентификатор любого состояния s_i построен по автомату A_M и не содержит переходов с предикатами. Соответственно последовательности такого идентификатора являются выполнимыми в любой конфигурации с состоянием s_i.

Таким образом, задача построения полного проверяющего теста относительно множества реализаций $\mathfrak{I}$ сводится к построению обхода графа переходов расширенного автомата-спецификации и добавления к этим последовательностям последовательностей соответствующих гармонизированных идентификаторов. Задача обхода графа переходов непосредственно связана с возможностью выполнимости входной последовательности в текущем состоянии расширенного автомата, что определяется значением соответствующего предиката. Для построения обхода графа переходов расширенного автомата мы используем специальный вид моделирования расширенного автомата, который, в общем случае, позволяет не строить в явном виде эквивалентный конечный автомат. Предлагаемое моделирование наследует идею метода [5] и осуществляется следующим образом. На начальном шаге полагаем список L пройденных переходов пустым. Строим дерево, например, глубины 2. Корню дерева ставится в соответствие начальная конфигурация расширенного автомата, а листьям – конфигурации, достигнутые из начальной вершины при подаче входных символов со всевозможными значениями входных параметров. Заносим в список L переходы, которые были пройдены при построении дерева. Объявляем финальными те вершины, которые помечены состояниями, из которых есть переходы, не попавшие в список L к настоящему моменту. Объявляем одну из финальных вершин корнем нового дерева, снова строим пути длины 2 и повторяем процедуру. Процедура заканчивается, если каждый переход автомата M занесен в список L или время, отведенное для моделирования, окончено. В последнем случае тест будет полным только относительно переходов автомата-спецификации M, которые занесены в список L.

Расширенный автомат спецификации может быть таким, что области определения контекстных переменных и входных параметров бесконечны или эквивалентный конечный автомат имеет огромное число состояний и переходов. В этих случаях можно использовать лишь частичное моделирование расширенного автомата. Суть такого моделирования заключается в том, что мы вводим определенные условия, при выполнении которых моделирование прекращается. Например, вводим ограничение на число состояний эквивалентного конечного автомата, и моделирование выполняется до тех пор, пока данная граница не достигнута.

Заключение

В данной работе предложен метод построения теста для расширенных автоматов, основанный на частичном моделировании расширенного автомата. Метод позволяет строить тесты с гарантированной полнотой, в том числе для случая, когда области определения контекстных переменных и входных параметров бесконечны. Для нахождения обхода графа переходов расширенного автомата используется метод частичного моделирования поведения автомата.

Одним из условий окончания моделирования поведения расширенного автомата может быть то, что иногда нам требуется протестировать лишь некоторую функцию протокола, описанного расширенным автоматом. В этом случае необходимо моделировать автомат не на всех входных последовательностях, а лишь на тех, которые отвечают за выполнение данной функции.

Конечный автомат спецификации получается не полностью определенным. Строим проверяющий тест для данного автомата относительно множества полностью определенных детерминированных реализаций (с числом состояний, например, не больше, чем у эталона) относительно квазиэквивалентности. Полнота теста гарантируется только в этом классе.

ЛИТЕРАТУРА

1. *Petrenko A., Boroday S., Groz R.* Confirming configurations in EFSM // IEEE TSE. – 2004. – V. 30. – No. 1. – P. 29 – 42.
2. *Chen W.-H.* Executable test sequences for the protocol data flow property // FORTE. – 2001. – P. 285 – 299.
3. *El-Fakih K., Prokopenko S., Yevtushenko N., Bochmann G.* Fault Diagnosis in Extended Finite State Machine // Lecture Notes in Computer Science. – 2003. – P. 197 – 210.
4. *Коломеец А.В., Прокопенко С.А.* Соответствие между ошибками в программных реализациях протоколов и расширенных автоматах // Вестник ТГУ. – 2005. – № 14. – С. 154 – 157.
5. *Cavalli A., Lee D., Rinderknecht C., Zaidi F.* Hit-or-jump: An algorithm for embedded testing with applications to in services // Proceedings of the Joint International Conference FORTE/PSTV '99. – IFIP TC6 WG6.1. – 1999.

УДК 519.710.20, 519.17:5

О ВОССТАНОВИМОСТИ ДИСКРЕТНЫХ МОДЕЛЕЙ ГЕННЫХ СЕТЕЙ

А.В. Комаров, А.А. Евдокимов, В.А. Лихошвай

Новосибирский государственный университет,
Институт математики СО РАН им. С.Л. Соболева,
Институт цитологии и генетики СО РАН

E-mail: medium@gorodok.net

В работе рассматривается дискретная модель генной сети, называемая АМ-автоматом, функционирование которой определяется заданием автоматного отображения на ориентированном графе. Найдено число экспериментов, необходимых и достаточных для восстановления АМ-автомата.

Ключевые слова: *генные сети, ориентированный граф, АМ-автомат.*

Введение

Генные сети являются сложными системами, состоящими из генов, РНК, белков, кодируемых соответствующими генами, а также других веществ и соединений [1]. Функционирование генной сети можно представить как изменение с течением времени концентраций веществ, входящих в её состав. Одним из важнейших свойств генных сетей является способность изменять концентрацию веществ сети в ответ на изменение условий внешней и внутренней среды. В связи с массовым секвенированием геномов резко возросла актуальность изучения закономерностей функционирования генных сетей в живых системах. В частности, стоит проблема восстановления структурно-функциональных связей в генных сетях по данным экспериментального анализа временных траекторий изменения концентраций веществ, синтезируемых в процессе функционирования генных сетей [2].

Рассматриваются задачи восстановления структуры графа сети по некоторым экспериментальным данным. В работе формализуется понятие эксперимента для модели АМ-автомата [3], находится число экспериментов, необходимых и достаточных для восстановления графа модели с ограниченной полустепенью захода вершин. Предложен эвристический подход для сужения множества допустимых моделей, учитывающий осуществимые на практике эксперименты, такие, как измерение экспрессии генов с помощью технологии MicroArray [4].

1. Основные определения

В модели АМ-автомата генная сеть представляется связным ориентированным графом $G(U, E)$ без петель, где U – множество вершин, соответствующих белкам, а E – множество дуг, имеющих смысл отрицательных регуляторных связей.

Пусть $U(G) = \{v_1,\dots,v_n\}$ – множество вершин ориентированного графа. Каждой вершине v_i приписывается целочисленный вес $x_i \in \{0,\dots, p-1\}$, $i=1\dots n$, который определяет концентрацию белка, а $p>0$ – верхняя гра-