

УДК 519.713

ББК 32.815

ВЛОЖЕННЫЕ АВТОМАТЫ, ИНЕРЦИОННАЯ МОДЕЛЬ КОНЕЧНЫХ АВТОМАТОВ

Любченко В.С.¹

(ФГБУН Институт проблем управления им.

В.А.Трапезникова РАН, Москва)

В работе описаны модель вложенных автоматов и базирующаяся на ней модель инерционного конечного автомата, содержащего переходы, которые, будучи запущены, могут не приводить к смене текущего состояния автомата. Показана эквивалентность данных моделей модели классического конечного автомата. На примерах рассмотрены свойства вложенных автоматов, актуальные для практики алгоритмизации дискретных моделей систем.

Ключевые слова: транспортная задержка, инерционная задержка, вложение автоматов, инерционный конечный автомат.

1. Введение

Расширение формальной модели преследует обычно цели расширения функциональных возможностей и/или повышения наглядности и удобства ее применения. Можно считать удачей, если они достигаются одновременно, и вдвойне, если при этом не нарушается свойство эквивалентности исходной модели. Последнее актуально для применимости существующих формальных методов анализа и синтеза базовой модели.

Рассмотренная далее формальная модель в полной мере отвечает обозначенным выше целям. С одной стороны, она обладает качествами, которые отсутствуют у базовой модели конеч-

¹ Любченко Вячеслав Селиверстович, ведущий инженер (slubch@mail.ru).

ного автомата, более наглядна и удобна для отражения определенных свойств реального объекта, и при этом не нарушает свойств базовой модели, оставаясь тем самым в рамках существующей теории, дополняя ее новыми возможностями.

Рассмотренный пример создания автоматной модели логического элемента задержки демонстрирует, с одной стороны, степень усложнения решения при необходимости детализировать свойства реального объекта на базе использования обычной модели. С другой стороны, он отражает качества рассматриваемой модели, которая в такой же ситуации порождает более компактные решения и реализует поставленные цели более наглядными и удобными средствами.

Кроме того, качества описанной далее модели вложенных автоматов позволяют говорить об определенном подклассе автоматных моделей – инерционных и соответствующем подклассе алгоритмов.

1. Постановка задачи

Пусть стоит задача создания дискретной модели логического элемента, реализующего без преобразования передачу значения входного сигнала на выход. Параметры сигналов, определяющие при этом значения задержек переключения элемента, показаны на временной диаграмме рис. 1.

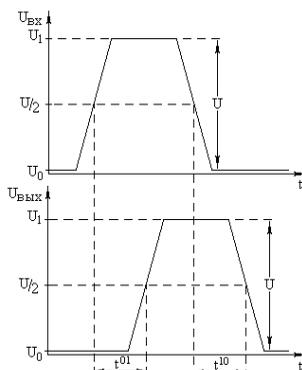


Рис. 1. Определение времени задержки логического элемента

Обычно времена задержек t^{01} и t^{10} , определяющие соответственно задержки переключения элемента в единичное и нулевое состояния, не совпадают. Кроме того, поскольку длительности фронтов сигналов меньше значений задержек, а также с учетом двоичной кодировки сигналов, за основу принимают следующее изображение и графическое представление задержек (по ГОСТ 2.743-91), показанное на рис.2.

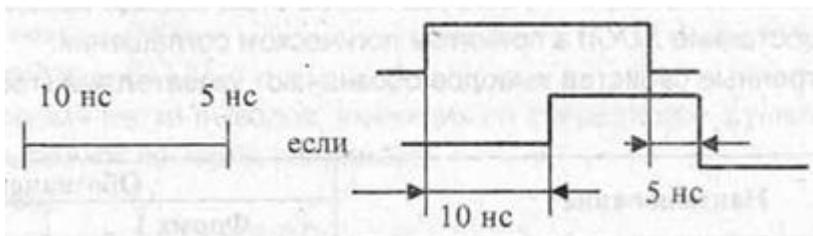


Рис. 2. Изображение задержки по ГОСТ 2.743-91

2. Автоматная модель задержки

Создавая модель в форме конечного автомата удобно моделировать значение задержек переключения выхода элемента числом дискретных тактов модели автомата *синхронного* типа (определение синхронных автоматов см. в [1]). Автоматная модель *единичной задержки*, времена переключений которой определяются единичным дискретным тактом синхронной модели, показана на рис.3 (для сравнения см. автоматную модель задержки, приведенную в [2]).

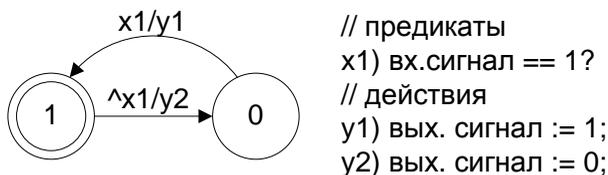


Рис. 3. Модель единичной задержки в форме автомата Мили

Приводя далее автоматные модели, будем иметь в виду, прежде всего, их *программную* реализацию, безусловно допуская и *аппаратную* реализацию моделей [3, 4]. При этом будем чаще всего опускать описание методов и свойств, например, объектно-ориентированной реализации модели (см. [8]), рассматривая в целях простоты только поведение объектов, ассоциируемое с автоматной моделью управления. В практике программирования реализация подобных элементов объектной модели не вызывает каких-либо проблем и в рамках дальнейшего обсуждения не требует подробных пояснений.

На рис.4 показаны две автоматные модели, так называемой, *транспортной задержки* [5] с моделированием значений t^{01} и t^{10} . Одна в форме автомата Мили (рис.4а), вторая – в форме совмещенной модели Мили-Мура (рис. 4б). В отличие от *инерционной задержки*, которая передает на выход лишь изменение сигналов длительностью не менее длительности задержек, транспортная задержка передает на выход любое изменение сигнала на входе [5, 6].

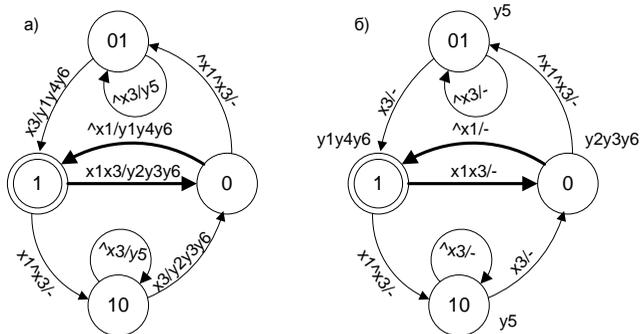


Рис. 4. Модели элемента с транспортными задержками в форме автоматов Мили(а) и Мили-Мура (б)

Для представленных моделей сигнал x_3 (при программной реализации ему соответствует одноименный *предикат* – функция без параметров, возвращающая булевское значение) принимает истинное значение при равенстве значения счетчика тактов модели значению переменной t , равной величине задержки

t^{01} или t^{10} . Это значение ей присваивается соответственно сигналами y_4 и y_3 (выходным сигналам поставлены в соответствие одноименные *действия* – программные функции, не имеющие в данном случае параметров и возвращаемого значения). Сигналы y_1 , y_2 присваивают значению свойству, поставленному в соответствие выходу элемента, y_5 – увеличивает счетчик тактов, который сбрасывается сигналом y_6 .

Модели на рис.4 отражают отличия в функционировании в целом эквивалентных моделей – автоматов Мили и Мура (совмещенная модель Мили-Мура объединяет в себе функционирование обоих типов моделей). Считается, что модель Мили гибче с точки зрения выдачи сигналов и в силу этого может содержать меньшее число состояний, но автомат Мура четче отражает функции внутренних состояний, ассоциированных с соответствующими сигналами.

В общем случае модель Мили удобнее, но в определенных ситуациях модель Мура надежнее, т.к. позволяет избежать ошибок, которые могут быть допущены при формировании переходов автоматов Мили. Так, для моделей на рис.4 при переходе от автомата Мили к совмещенной модели проявились нюансы, связанные с подсчетом тактов. Результатом применения «идей Мура» стала более корректная работа со счетчиком тактов, реализуемая в цикле сигналом y_5 . Дополнительно модель упростилась за счет исключения повторов записи сигналов на переходах модели Мили.

3. Вложенная модель автоматов

Реализация подсчета тактов в дополнительных состояниях, моделирующих задержки элемента, усложнило формальную модель (ср. рис.3, рис.4). Неизбежное в такой ситуации введение дополнительных внутренних состояний привело фактически к утрате одного полезного свойства модели – ассоциации состояний модели с состояниями ее выхода. Это привело к необходимости **обязательного** введения переменной и сигналов, задающих ее значение. Заметим, что для модели на рис.3 сигналы y_1 и y_2 лишь, скорее, отражали подобную возможность, чем были необходимостью.

Однако, используя механизм вложения моделей, аналогичный механизму обычных подпрограмм, можно фактически привести вид сложной модели на рис.4 к ее исходному виду на рис.3. Подобная модель, использующая вложение автоматов, показана на рис.5. В ее рамках вложенный автомат реализует транспортную задержку, подсчитывая дискретные такты, и возвращает управление модели верхнего уровня при равенстве счетчика тактов значению задержки.

С использованием механизма вложения автоматов модель на верхнем уровне иерархии рис.5а по структуре совпадает с моделью на рис.1. При этом моделирующий задержки вложенный автомат на рис.5б также представляет собой гораздо более простой автомат по сравнению с одноуровневой моделью на рис.4. Более того, как «библиотечный автомат», он может быть использован в любой другой автоматной модели.

Замечание 1. В целях упрощения графического представления автоматной модели далее будет использован прием замены нескольких дуг переходов, принадлежащих общим состояниям, одной дугой.

На рис.5а этому соответствуют две дуги, показанные пунктирными линиями, которые заменены одной дугой, изображенной утолщенной линией (этому соответствует замена переходов $x_1x_3/-$ и $x_1^2x_3/z_1$ переходом x_1/z_1). При программной реализации подобная замена реализуется в действии, создающем вложенный автомат (z_1). В результате [программная] реализация автомата фактически в точности соответствует его графической форме описания.

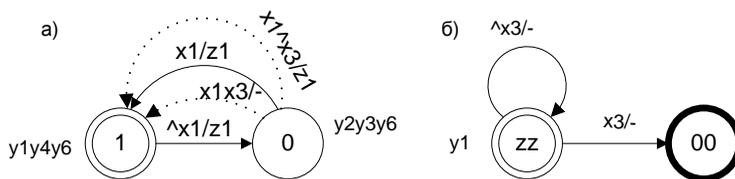


Рис. 5. Модель элемента(а) с транспортными задержками (б) в форме вложенных автоматов

Рассматривая иерархическую модель элемента задержки необходимо, во-первых, обратить внимание на сигнал с именем $z1$, который реализует механизм вложения автоматов, и, во-вторых, на наличие у вложенного автомата заключительного состояния с именем «00». Ни чем другим по формальным признакам вложенная модель не отличается от модели верхнего уровня.

Завершая свою работу, вложенный автомата переходит в состояние «00», определяя тем самым возврат на предыдущий уровень иерархии. Ни каких ограничений на уровень вложения (*ранг* вложенности) при этом не накладывается. Вложение автоматов позволяет также реализовать механизм *рекурсивного вложения автоматов*.

4. Модель инерционного автомата

Логический элемент, реализующий принцип инерционной задержки, более адекватен реальным объектам, которые, как известно, характеризуются определенной *полосой пропускания*, не позволяющей им реагировать на слишком быстрые преобразования входных сигналов[1]. Модель элемента инерционной задержки в форме совмещенной автоматной модели Мили-Мура показана на рис.6.

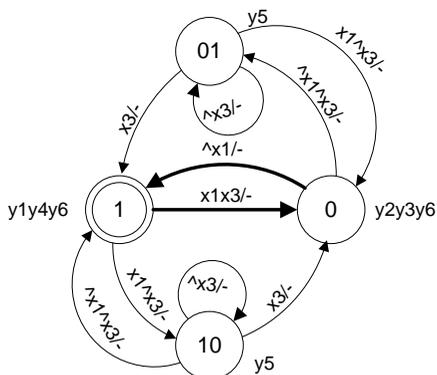


Рис. 6. Автоматная модель элемента с инерционными задержками

Из сравнения данной модели с моделью на рис.4б очевидно их достаточно небольшое структурное различие, которое сводится к возврату из переходных состояний в соответствующее внутренне состояние «0» или «1», если значение сигнала восстановилось до истечения времени задержки. Однако, даже такое незначительное отличие оказывает качественное влияние, не позволяя использовать отработанный механизм вложения автоматов.

Замечание 2. В существующей алгоритмической модели после выхода из подпрограммы вычисления возобновляются со следующей после вызова подпрограммы команды/инструкции.

Для решения возникшей проблемы предлагается модель вложения (модель подпрограмм), которая не исключает возврат алгоритма в точку, предшествующую вызову подпрограммы. При внимательном анализе можно видеть, что подобный иерархический механизм вложения моделей позволяет создавать новые по качеству и смыслу алгоритмы.

Реализовать предлагаемую модель вложенных автоматов можно, если для вложенного автомата, кроме его нормального состояния завершения с именем «00», ввести еще одно завершающее состояние с именем, например, «XX». Переход в него при завершении работы вложенного автомата оставляет автомат верхнего уровня в том же состоянии, переход из которого привел к вызову вложенного автомата.

Модель, включающая подобные вложенные автоматы (с двумя видами заключительных состояний), показана на рис.7. Чтобы отличить переходы автомата, которые будучи запущенными на исполнение, могут не привести к изменению текущего состояния, от обычных переходов будем их называть *инерционными переходами* и изображать пунктирной линией. Модель автомата, содержащую подобные переходы, будем называть *инерционным конечным автоматом* или просто *инерционным автоматом*.

Теперь на верхнем уровне представления лишь наличие инерционных переходов отличает модель на рис.7а от модели на рис.5а. При этом разницу в поведении моделей определяют, как и прежде, вложенные автоматы, которые в случае инерционной задержки содержат два вида заключительных состояний –

обычное с именем «00» и инерционное заключительное состояние с именем «XX».

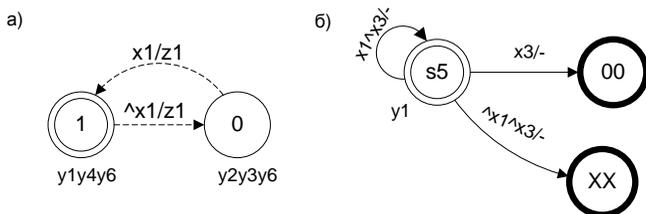


Рис. 7. Модель элемента (а) с инерционными задержками (б) в форме вложенных автоматов

Введение нового типа автомата – инерционного определяется трактовкой функционирования автоматной модели: инерционный автомат имеет переходы, запуск которых определяет процедуру *отложенного перехода* в новое состояние. Подобных переходов у обычного автомата нет. У последнего сработавший переход **обязательно** (!) приводит к смене текущего состояния (в случае петли оно просто совпадает с текущим состоянием).

5. Эквивалентность моделей

У инерционного автомата любой его переход инерционного типа порождает вложенный автомат. Последний в обязательном порядке содержит заключительное состояние «XX» и, соответственно, переход в него. Тем не менее, все это не влияет на общие свойства модели, которая подобно любой другой модели, содержащей вложенный автомат, эквивалентны базовой модели конечного автомата.

На рис.8 показана процедура «раскрытия» перехода, содержащего вложенный автомат, приводящая любой иерархический автомат к эквивалентной одноуровневой модели. Для рассматриваемой модели в этом случае были, заметим, восстановлены два замещенных одной дугой перехода (см. также рис.5а), среди которых «раскрытие» выполнено для перехода $x1^x3/z1$.

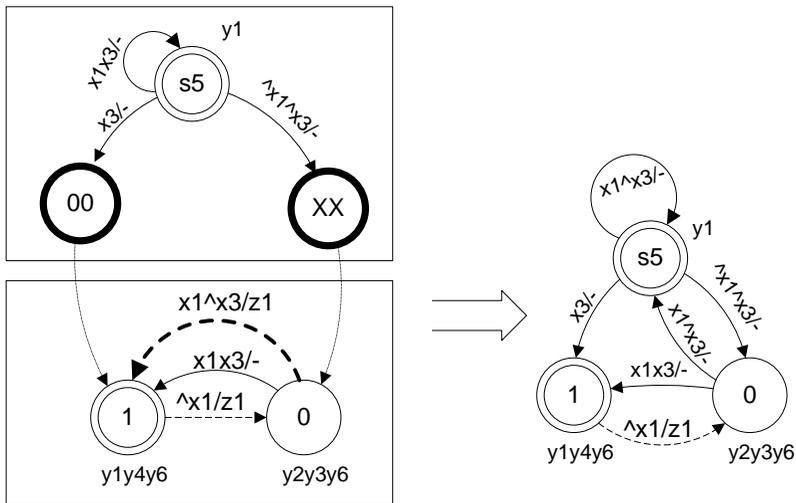


Рис. 8. Приведение вложенного автомата с одноуровневой автоматной модели

Легко видеть, что любая из вложенных моделей путем подобных обратных преобразований приводится к эквивалентному обычному автомату. При раскрытии вложенного автомата новое текущее состояние автомата верхнего уровня становится начальным состоянием вложенного автомата, заключительное состояние последнего с именем «00» совмещается с состоянием, в которое должен перейти автомат верхнего уровня, а заключительное состояние с именем «XX» совмещается с состоянием, из которого выходит рассматриваемый переход автомата верхнего уровня.

Однозначность рассмотренных преобразований доказывает, что функционирование любых автоматов, имеющих вложенные автоматы рассмотренных выше видов, эквивалентно функционированию соответствующих им обычных одноуровневых конечных автоматов.

6. Заключение

Чем модель проще, чем меньше она загромождена мелкими особенностями, тем она яснее, понятнее и, как следствие, надежней. Но, безусловно, подобные упрощения не должны вести к потере качества – способности точно представлять будь то алгоритм поведения моделируемого объекта или некий вычислительный алгоритм. Модульность и иерархичность, вводимая механизмом вложения моделей (в программировании – этому соответствует механизм подпрограмм), во многом служит именно этим задачам.

Часто модификации моделей проводятся под лозунгом их улучшения. Например, с точки зрения практики. Для автоматных моделей примером этого могут служить автоматы с входными сигналами в форме неких событий – событийные и им подобные модели; появление неименованных (?!) состояний; вложенных состояний и т.д. и т.п. (см. определение автоматов в UML [7]). А это не очень хорошо, т.к. фактически исключает использование так необходимых при проектировании формальных методов теории базовой модели.

В работе представлен механизм создания иерархической автоматной модели, порождающий, в том числе, новую модель конечного автомата, названную инерционной. Вложение автоматов и модель, отражающая инерционные свойства реальных объектов и процессов, позволяют создать технологию проектирования, не нарушая при этом базовых свойств формальной модели и, соответственно, не выводя проектирование за рамки базовой теории автоматов.

Технология проектирования программных систем, базирующая на автоматной модели управления и реализующая описанные выше возможности, представлена в рамках среды ВКПа[8]. Результаты тестирования представленных выше моделей транспортной и инерционной элементов задержек в ВКПа показаны на рис.9.

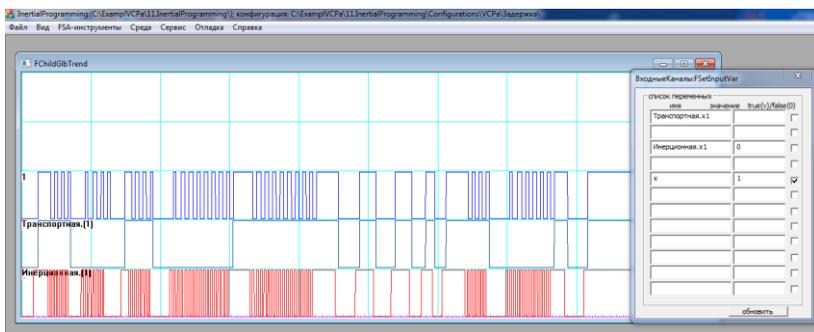


Рис. 9. Тестирование моделей задержек в среде ВКПа.

Легко видеть, что инерционная задержка адекватно отражает свойство реальных объектов устранять «дребезг» слишком быстрого изменения входного сигнала. При этом, повторимся, на верхнем уровне иерархии, как было показано в работе, модель, реализующая достаточно сложный алгоритм обработки «инерционных сигналов», почти не отличается по форме от ее самого простого варианта.

Литература

1. ГЛУШКОВ В.М. *Введение в кибернетику*. Изд-во АН Украинской ССР. К.: 1964. - 324с.
2. КУДРЯВЦЕВ В.Б., АЛЕШИН С.В., ПОДКОЛЗИН А.С. *Введение в теорию автоматов* – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 320 с.
3. МАЙОРОВ С.А., НОВИКОВ Г.И. *Принципы организации цифровых машин*. – Л.: Машиностроение, 1974. – 431 с.
4. БАРАНОВ С.И. *Синтез микропрограммных автоматов*. – Л.: Энергия, 1979. – 232с.
5. АРМСТРОНГ ДЖ.Р. *Моделирование цифровых систем на языке VHDL*: Пер с англ./М.: Мир, 1992. – 175 с.
6. МИЛЛЕР Р. *Теория переключательных схем, т.2 – Последовательностные схемы и машины* : Пер. с англ. – М.: Наука, 1971. – 304с.

7. РАМБО ДЖ., ЯКОБСОН А., БУЧ Г. *Язык UML. Руководство пользователя. Второе издание.* Академия АЙТИ: Москва, 2007. – 493 с.
8. ЛЮБЧЕНКО В.С. *Автоматная парадигма параллельного программирования //Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск).* - М.: Изд-во МГУ, 2012. - 752 с. ISBN 978-5-211-06394-5.

THE ENCLOSED FINITE-STATE MACHINES, INERTIAL MODEL OF THE FINITE-STATE MACHINES

Viacheslav Lyubchenko, Institute of Control Sciences of RAS, Moscow, principal engineer (sllubch@mail.ru).

Abstract: In work models of the enclosed automatic machines and model of the inertial final automatic machine containing transitions based on it which, being are started are described, can not lead to change of a current condition of the automatic machine. Equivalence of the given models of model of the classical final automatic machine is shown. On examples properties of the enclosed automatic machines, actual algorithmizations for practice of discrete models of systems are shown.

Keywords: Transport delay, inertial delay, investment of automatic machines, the inertial final automatic machine.

Статья представлена к публикации
членом редакционной коллегии ...заполняется редактором...

Поступила в редакцию ...заполняется редактором...
Опубликована ...заполняется редактором...