

УДК 519.713

ББК 32.815

ДИЗЬЮНКТИВНЫЕ ВЛОЖЕННЫЕ АВТОМАТЫ, ИНЕРЦИОННАЯ МОДЕЛЬ КОНЕЧНЫХ АВТОМАТОВ

Любченко В.С.¹

(ФГБУН Институт проблем управления им.

В.А.Трапезникова РАН, Москва)

В работе дано описание модели дизьюнктивных автоматов и созданных на их базе модели вложенных автоматов, которые являются основой новой модели инерционного конечного автомата. Доказана эквивалентность введенных моделей классическому конечному автомату. На примерах рассмотрены свойства введенных моделей, актуальные для практики алгоритмизации дискретных моделей систем.

Ключевые слова: автомат с абстрактным состоянием, дизьюнктивный автомат, вложенный автомат, инерционный автомат, модель элемента задержки.

1. Введение

При совершенствовании формальной модели важно сохранить эквивалентность базовой модели. Иначе невозможна преимущество теории, без которой на практике будет сложно. В подобной ситуации оказалась модель автоматов в языке UML[13]. Здесь конечные автоматы и классический конечный автомат (см. [8]) кроме названия уже мало что связывает, а потому о применении в UML теории автоматов можно говорить весьма условно.

Иной подход демонстрирует формулировка модели структурированного автомата, описанная в [1, 2]. В этом случае, наряду с существенной корректировкой автоматной модели

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

(заимствующей достоинства модели технологии структурного проектирования программ), постарались сохранить максимальную совместимость с базовой моделью. И вызывает только сожаление, что данная модель не получила в последующем развития. Но сам подход авторов и механизм совершенствования модели достойны того, чтобы быть отмеченными.

Но если вернуться к UML или даже более широко – к проектированию программных систем, то на текущий момент простая, понятная, удобная и безупречная теоретически классическая модель автомата здесь не востребована. А потому, как можно предположить, для практики UML она фактически потеряна, в результате чего фактически растеряны и связи с ее теорией.

Безусловно, есть области, где модель автомата в классическом определении и соответствующая ей теория востребованы. Но они достаточно специфичны. Это, прежде всего, теория языков, проектирование цифровых схем и т.п. Для универсального применения автоматам требуются качественные изменения. И это весьма актуально, т.к. современной практике требуются формальные модели вполне конкретного качества. Особенно, например, параллельные.

Существующие попытки, что-то изменить в модели конечного автомата и его теории (см. подробнее [7]) явно не поспевают за «нововведениями» в том же языке UML. На текущий момент конечные автоматы в UML и классические конечные автоматы это фактически разные модели, но признание этого должно стать поводом для внесения необходимых изменений в обе модели, сохранив достоинства каждой из них.

Для решения поставленных целей необходима, видимо, более совершенная модель конечного автомата. Она должна учитывать, с одной стороны, структуру входных и выходных каналов систем, а, с другой, подобно абстрактной модели абстрагироваться от реализации понятия внутреннего состояния.

Хотелось бы, чтобы упрощения модели касались не только ее самой, но и ее реализации. В современных условиях это особенно актуально для случая с ее программированием. В дальнейшем подобная модель должна стать основой для струк-

турно более гибкой и адекватной реальным системам параллельной сетевой модели. При этом важно, чтобы все нововведения позволяли использовать теорию абстрактных автоматов.

Описанное далее «структурно-абстрактное» расширение автоматной модели, преследуя заявленные выше цели, является в определенной степени компромиссом понятий, представленных на структурном уровне в [7, 10], а на абстрактном в [6, 11]. Такая «компиляция» позволяет на абстрактном уровне учитывать структуру каналов реальных объектов, а структурной теории – не отвлекаться на реализацию внутреннего состояния.

Научная ценность и актуальность представленной в работе модели определяется тем, что она в полной мере отвечает обозначенным выше целям. Введенная модель автомата более функциональна, универсальна, наглядна и компактна, а потому более удобна для моделирования и отражения свойств реальных объектов. В то же время она не нарушает основ базовой модели, позволяя тем самым оставаться в рамках существующей теории, дополняя ее при этом новыми возможностями.

Качества предлагаемых в работе моделей позволяют в конечном итоге выделить новый подкласс формальных моделей – инерционных автоматов или, что более общо, – подкласс инерционных алгоритмов.

Рассмотренный пример модели логического элемента отражает качества новых моделей, которые порождают более компактные и ясные решения. Но при этом сам выбор того или иного варианта модели автомата остается за разработчиком.

Для понимания описанной модели автомата нам потребуются понятия и теоремы, не выходящие в целом за область начал теории *булевых функций (БФ)* (более подробно о теории БФ см. [6, 10] или [14])

2. Базовые понятия и определения

Можно утверждать, что фактически у любой теории есть проблемы, которые становятся очевидными на практике. Не являются исключением в этом смысле и автоматы.

Так, ограничение на число входов абстрактной модели конечного автомата (одним входным и одним выходным каналом)

с практической точки зрения можно оправдать только удобствами самой теории да спецификой прикладных областей, использующих формальную модель в чистом виде. Можно видеть, что и теория структурных автоматов не дает ответы на ряд актуальных практических проблем многокомпонентных систем.

Безусловно, серьезные претензии можно предъявить и к практике, которая, порой, так «улучшает» модели, что об осмысленной их теории, порой, речи не идет и идти не может. К этому привела, как можно видеть, практика языка UML.

Поэтому подобные модели правильнее считать уже иной формальной моделью, которую, возможно, и роднит с исходными конечными автоматами общая терминология – конечный автомат, внутренние состояния, переходы и т.п.

2.1. АВТОМАТ С АБСТРАКТНЫМ СОСТОЯНИЕМ

*Определение 1. Назовем **автоматом с абстрактным состоянием** конечный автомат (см. также [6]), у которого каналы представлены множеством входных – $x_1, x_2, \dots, x_n$ и выходных булевых переменными – $y_1, y_2, \dots, y_m$, а внутреннее состояние одной многозначной переменной q . Его **функция переходов** определяет следующее состояние $q(t+1)$ ¹ в зависимости от текущего значения состояния q и значения двоичной входной переменной, представленной булевой функцией – $f(x_1, x_2, \dots, x_n)$ от входных переменных в дизъюнктивно нормальной форме (ДНФ). **Функция выходов** автомата определяет зависимость последующего значения выходной переменной $y(t+1)$, представленного **элементарной конъюнкцией** – $(y_1, y_2, \dots, y_n)$ выходных булевых переменных без знаков отрицания:*

$$(1) \quad q(t+1) = \varphi(q(t), x(t)),$$

$$y(t+1) = \psi(q(t), x(t)),$$

где $x = f(x_1, x_2, \dots, x_n)$, а $y = (y_1, y_2, \dots, y_n)$.

Автоматы, определяемые уравнениями (1), будем называть *автоматами Мили*.

Замечание 1. Привязка изменения выходных каналов автомата к **следующему** моменту времени (подобно изменению

¹ Иногда пишут q^+ [6]

состояния) исключает мгновенную реакцию на изменение входной информации. Такой подход отражает свойство **инерционности**, присущее реальным системам.

В существующей теории автоматов входной и выходной сигналы привязаны к одному и тому же моменту времени – t . Из-за этого и появляется их «мгновенная» связь. Иногда, правда, признается, что выходной сигнал $y(t)$ реального автомата всегда появляется после входного сигнала $x(t)$, но, опять же, только в привязке к состоянию $q(t)$ или $q(t-1)$. При этом автоматы называют *правильными*, если выходной сигнал не зависит от входного, а определяется лишь его состоянием (подробнее см. [11]).

Замечание 2. Закон функционирования автоматов, который не привязывает изменение выходного сигнала к текущему времени, исключает необходимость различать, как это принято (см. [11]), обычные и **сдвинутые** функции выходов.

Изменение выходного сигнала, привязанное к следующему моменту времени, вводит в определенном смысле еще один вид автоматов, которые, образно говоря, правильнее неправильных и даже правильных классических конечных автоматов. И, что примечательно, пожалуй, *только у автоматов, изменив лишь закон их функционирования, можно качественно повлиять на свойства формальной модели.*

Замечание 3. Исключение **мгновенной** реакции выхода решает одну из проблем, которая не имеет решения в рамках классических понятий теории автоматов. К ним относится проблема корректности схем, включающих **циклические цепи** [5].

Определение 2. Назовем **автоматом Мура** автомат, у которого функции переходов и выходов следующие:

$$(2) \quad \begin{aligned} q(t+1) &= \varphi(q(t), x(t)), \\ y(t+1) &= \psi(q(t+1)). \end{aligned}$$

У автоматов Мура значение выходной переменной определяется значением только переменной внутреннего состояния (или просто внутренним состоянием). Подобные автоматы, как уже было сказано выше, называются правильными.

Замечание 4. В существующей теории автоматами Мура называют правильные автоматы, выходной сигнал которых зависит от $q(t)$, а не от $q(t-1)$.

Определение 3. Назовем **совмещенным автоматом Мили-Мура** автомат, функции переходов и выходов которого определяются следующим образом:

$$\begin{aligned}(3) \quad & q(t+1) = \varphi(q(t), x(t)), \\ & y_1(t+1) = \psi_1(q(t), x(t)), \\ & y_2(t+1) = \psi_2(q(t+1)).\end{aligned}$$

Данная модель совмещает в себе функции автоматов Мили, которым соответствует функция выходов $y_1(t+1)$, и Мура – $y_2(t+1)$.

Замечание 5. В совмещенной модели проявляется отличие в поведении моделей: автомат Мили выдает выходные сигналы **до** момента изменения текущего состояния, а автомат Мура – **после** его изменения. Но и то и другое должно быть завершено до наступления следующего момента дискретного времени.

Безусловно, реализуя функциональность модель, необходимо на текущем дискретном такте обеспечить завершение всех автоматных операций, связанных с выдачей сигналов и изменением текущего состояния до момента наступления очередного дискретного такта автоматного времени (в практике цифровых схем это реализуется подбором частоты генератора синхросигналов).

2.2. ДИЗЬЮНКТИВНЫЕ АВТОМАТЫ

Определение 4. Назовем **автоматом в дизьюнктивной нормальной форме (ДНФ КА)** или просто **дизьюнктивным автоматом** автомат $A = (X, Q, Y, q_1, F(x \in X/y \in Y))$, где:

X – множество переменных векторного пространства конечного множества входных булевских переменных $x_1, x_2, \dots, x_m$, множество X^* элементов векторного пространства которого представляет входной алфавит автомата;

Q – конечное непустое множество (состояний);

Y – множество переменных векторного пространства конечного множества выходных булевских переменных $y_1, y_2, \dots, y_n$, подмножество элементов Y^* векторного пространства которого – выходной алфавит автомата;

$q_1 \in Q$ – начальное состояние;

$F(x \in X/y \in Y)$ – отображение множества Q в себя, определяющее функции переходов и выходов автомата.

Отображение F представляет отображение множества Q в себя, которое любому $q \in Q$ и каждой входной букве $x \in X$ сопоставляет состояние $q_k \in Q$, определяющее функцию переходов $\varphi(q, x)$, и выходную букву $y \in Y$, определяющую функции выходов $\psi(q, x)$ для автоматов Мили, Мура или их совмещенной модели Мили-Мура (для сравнения см. определение абстрактного автомата в [11]).

В отображении F дизъюнктивного автомата (далее просто автомата, если не оговорено иное) входные буквы, определяющие условие перехода в новое состояние, представлены ДНФ булевой функцией от m входных булевых переменных, называемых также входными сигналами автомата, а выходные буквы, называемые также выходными сигналами, – элементарными конъюнкциями выходных булевых переменных без знаков отрицания.

Для отображения F допускается наличие безусловных переходов и переходов без выдачи выходных сигналов. Ситуация *безусловного перехода* соответствует пустому входному символу и будет обозначаться прочерком на месте условия перехода в следующее состояние в функции переходов. Любое состояние может содержать *один и только один* переход безусловного типа. Отсутствие выдачи выходных сигналов в рамках отображения F – *пустой выходной символ* будет изображаться прочерком или просто отсутствием символов на месте выходного сигнала в функции выходов.

Условие перехода в новое текущее состояние в случае ДНФ КА может быть представлено как отдельными булевскими переменными, так и элементарными конъюнкциями или функциями в форме ДНФ. Примеры правильного задания: $x5, -, x1 \wedge x4x6, \wedge x1V \wedge x2V \wedge x1x5$. Неправильного – $x1(\wedge x4Vx6)$. Примеры определения правильных выходных сигналов (символов выходного алфавита) – $y1y5, y15, -, y2y8y9y25$.

Замечание 6. Любой автомат с абстрактным состоянием, включающий произвольную аналитическую запись булевой

функции в качестве условия перехода, может быть преобразован в эквивалентную форму ДНФ КА.

В основе отмеченного преобразования известное из теории БФ доказательство нахождения ДНФ для любой БФ.

По аналогии с обычными автоматами в зависимости от области определения функции, представленной элементами векторного пространства входных переменных, рассматриваются полностью определенные и частичные автоматы. Но далее, если не указано иное, во внимание будут приниматься только полностью (*вполне*) определенные автоматы.

Заметим, что при этом выходной алфавит составляет, как правило, подмножество элементов векторного пространства выходных переменных.

Определение 5. Назовем автоматом в совершенной дизъюнктивной нормальной форме (СДНФ КА) автомат, условия переходов которого содержат только конститuentы единиц, т.е. элементарные конъюнкции, включающие булевские переменных всех входных каналов автомата, или булевскую функцию в СДНФ.

Лемма 1. Любой ДНФ КА представлен эквивалентным автоматом в СДНФ КА единственно.

Доказательство леммы 1 основывается на известном факте из теории БФ, о единственности представления любой булевой функции ее СДНФ.

Рассмотрим пример описания автомата в дизъюнктивной форме, представленный аналитической формой описания автоматов (подробнее о способах описания автоматов см. [11]).

Пример 1. Пусть задан дизъюнктивный автомат $A = (X, Q, Y, q_1, F(x \in X/y \in Y))$, где $X = \{x_1, x_2\}$, $Q = \{q_0, q_1\}$, $Y = \{y_0, y_1\}$, а отображение A определяется следующим образом:

$$Fq_0 = \{q_0(x_1 \wedge x_2 \vee x_1 x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\},$$

$$Fq_1 = \{q_0(x_1 \wedge x_2 \vee x_1 x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\}.$$

По виду отображения F можно сразу сказать, что автомат A имеет два входных – x_1, x_2 и два выходных канала – y_0, y_1 и представляет совершенную форму дизъюнктивного автомата – СДНФ КА.

Для минимизации описания булевых функций часто используется понятие *интервального* представления пространства

булевых переменных[6]. В его рамках отдельный терм покрывает интервал – множество термов, представляющих конstituенты единиц (нулей). На интервальном представлении базируется минимизация булевых функций, а на восстановлении интервалов – нахождение СДНФ булевой функции.

Эквивалентную для СДНФ КА дизъюнктивную форму автомата – ДНФ КА можно получить, выполнив «интервальную минимизацию» булевых функций. Например, отображение F для примера 1 в этом случае примет следующий вид:

$$Fq0 = \{q0(x1 \vee x2/y0), q1(\wedge x1 \wedge x2/y1)\},$$

$$Fq1 = \{q0(x1 \vee x2/y0), q1(\wedge x1 \wedge x2/y1)\}.$$

Замечание 7. Минимизация переходов СДНФ КА приводит в общем случае к эквивалентному автомату в форме ДНФ КА.

Граф автомата A из примера 1 показан на рис. 1а. Граф автомата, демонстрирующий «дизъюнктивный» характер переходов модели ДНФ КА путем «расщепления» перехода из состояния $q1$ в $q0$ по количеству термов БФ, показан на рис. 1б.

Обратная «расщеплению» операция «склейки» переходов демонстрирует более компактное описание дизъюнктивных автоматов за счет переходов, соединяющих попарно одинаковые состояния, помеченных одинаковыми выходными сигналами в форме одной БФ.

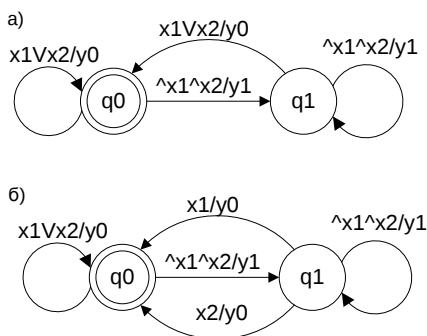


Рис. 1. Графы автоматов для примера 1.

Но, заметим, часто более ясное представление о работе модели дает именно «расщепленное» описание переходов. При этом важно отслеживать условие корректности подобных пере-

ходов, которые формально нарушают свойство ортогональности переходов автоматной модели.

Теорема 1. Для обеспечения корректности ДНФ КА, имеющего не ортогональные переходы, необходимо и достаточно, чтобы подобные переходы, имеющие общее исходное состояние, имели попарно одно и то же целевое состояние и были помечены одинаковыми выходными сигналами.

Доказательство теоремы базируется на операции «склейки» переходов, которая приводит к переходу, помеченному ДНФ булевой функции. Ее СДНФ определяет уже множество ортогональных переходов, помеченных ее отдельными термами (конституентами единиц).

Автомат А из примера 1 представляет автомат Мили. Эквивалентный ему автомат Мура показан на рис.2а. Пример эквивалентного им обоим совмещенного автомата Мили-Мура показан на рис.2б.

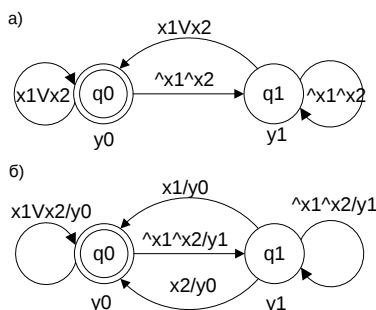


Рис. 2. Графы моделей автоматов Мура(а) и Мили-Мура (б) для примера 1.

Аналитическое описание совмещенной модели автомата для автомата на рис.2б будет иметь следующий вид:

$$(3) \quad \begin{aligned} Fq_0(y_0) &= \{q_0(x_1 \vee x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\}, \\ Fq_1(y_1) &= \{q_0(x_1 \vee x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\}, \end{aligned}$$

где состоянию q_0 поставлен в соответствие сигнал y_0 , q_1 – y_1 .

Далее будет использоваться чаще форма аналитического описания автоматов, которая для совмещенной модели Мили-Мура будет иметь следующий вид:

Е:

$$q_0(y_0) = \{q_0(x_1 \vee x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\}, \\ q_1(y_1) = \{q_0(x_1 \vee x_2 / y_0), q_1(\wedge x_1 \wedge x_2 / y_1)\}.$$

Как можно видеть, дизъюнктивные автоматы, используя интервальное свойство элементарных конъюнкций, имеют в сравнении с обычными автоматами в общем случае более компактную форму. Причем, как показывает практика, чем больше число входных каналов имеется у модели, тем более компактно будет ее интервальное описание.

2.3. ХАРАКТЕРИСТИЧЕСКИЕ АВТОМАТЫ

Создание полностью определенного автомата задача, трудоемкость и сложность которой резко увеличивается с увеличением числа входных каналов. Уменьшить ее можно, используя методы компактного представления множеств.

Для ДНФ КА это можно сделать, используя понятие *характеристического множества* пространства входных булевых переменных. Для БФ в этом случае в расчет берутся значения функции, на которых она принимает определенное значение. Так, характеристическое множество при табличном способе описания БФ может быть представлено строками, на которых она принимает единичное значение. Результатом этого является дизъюнктивная форма булевых функций.

Замечание 8. Описание полностью определенного ДНФ КА в общем случае более громоздко в сравнении с описанием БФ, т.к. автомат даже для отдельного своего состояния включает переходы, определенные на **всем** пространстве входных булевских переменных.

Покажем, что самая существенная минимизация описания конечных автоматов в общем случае достигается на базе идеи характеристического множества.

Определение 6. Будем называть *характеристической формой ДНФ КА* или просто *характеристическим дизъюнктивным автоматом* – **ХДНФ КА** автомат, включающий только переходы, изменяющие существенно общее состояние автомата. Множество исключенных переходов будем называть *дизъюнктивным дополнением до вполне определенного дизъюнктивного автомата* или просто *дополнением дизъюнктивного автомата* – **ДДНФ КА**. Для совершенных форм

это будут, соответственно, сокращенно – СХДНФ КА и СДДНФ КА.

Здесь под общим состоянием автомата понимается внутреннее состояние автомата и/или состояние его выходных каналов. Если рассматривать графическую форму описания автомата, то исключенными переходами будут петли с прочерками на месте выходных сигналов.

Если рассматривать произвольный полностью определенный автомат A^* , то:

$$(4) \quad A^* = A \cup A^c,$$

где A – подавтомат в форме ХДНФ КА, а A^c – дополнение автомата A до A^* .

Таким образом, любой полностью определенный автомат в дизъюнктивной форме является объединением двух автоматов в характеристической форме. Один из них отражает поведенческую сторону конечного автомата, другой – пассивную. Но, заметим, даже в пассивном состоянии автомат продолжает анализировать входную ситуацию, чтобы оперативно отреагировать на любое ее изменение.

Лемма 2. *Любой полностью определенный ДНФ КА может быть представлен объединением двух подавтоматов СХДНФ КА и СДДНФ КА единственно.*

Доказательство данной леммы основано на единственности представления в СДНФ любой булевой функции, а также на определении характеристического множества, разбивающего множество элементов векторного пространства на подмножества без пересечений.

Следствием леммы 2 является однозначность представления любого ДНФ КА по его «характеристическому подавтомату» в форме ХСДНФ КА.

Теорема 2. *Любой полностью определенный автомат в форме ДНФ КА представляется автоматом в форме ХСДНФ КА единственно.*

Если задан подавтомат в форме ХСДНФ КА, то найти его дополнение СДДНФ КА не представляет проблем. Имея СДНФ функций переходов, легко определить элементы характеристического множества, которые трактуются как петли при состояниях с прочерком на месте выходных сигналами.

Обратное, т.е. восстановление подавтомата СХДНФ КА по подавтомату в форме СДДНФ КА невозможно. В подобной ситуации идентифицируется лишь множество ситуаций, вызывающих ситуации переходов, и не более того. В какие состояния ведут данные переходы, какие при этом выдаются сигналы восстановить невозможно.

Для совмещенной модели примера 1 представление отображения F в форме ХДНФ КА будет следующим:

$$(5) \quad F: \\ q_0(y_0) = \{q_1(\wedge x_1 \wedge x_2)\}, \\ q_1(y_1) = \{q_0(x_1 \vee x_2)\}.$$

В данном описании автомата (ср. с (3)) исключено дублирование сигналов, которые на переходах автомата Мили в состояния, которые имеют такой же ассоциируемый с целевым состоянием выходной сигнал. Под минимизацию попали петли, выдающие сигналы, совпадающие с сигналами состояний.

Дополнение автомата $\wedge F$ до полностью определенного автомата F^* будет иметь следующий вид:

$$\wedge F: \\ q_0 = \{q_0(x_1 \vee x_2)\}, \\ q_1 = \{q_1(\wedge x_1 \wedge x_2)\}.$$

Поскольку данный подавтомат в соответствии с определением не изменяет общее состояние автомата, то из описания исключены сигналы, ассоциированные с внутренними состояниями по типу автоматов Мура.

Подчеркнем, что минимизации в совершенной форме описания автоматов подвергаются булевы функции, составленные из дизъюнкций булевых функций переходов, соединяющих попарно одни и те же состояния и помеченные одинаковыми выходными сигналами.

Теорема 3. *Любой полностью определенный ДНФ КА однозначно представляется дизъюнктивным автоматом в характеристической форме – ХДНФ КА.*

Доказательство теоремы основывается на единственности представления любой ДНФ функции в СДНФ и теореме 1.

Далее именно ХДНФ КА мы будем называть, если это не вносит путаницу, дизъюнктивным автоматом или просто автоматом.

По описанию в форме ХДНФ КА, если это необходимо, всегда можно найти полностью определенный дизъюнктивный автомат и/или его дополнение до данного автомата.

3. Вложенные автоматы

С алгоритмической точки зрения модель автоматов подобна программам без подпрограмм: среди конечных автоматов нет модели, имеющей структурную иерархию, аналогичную вложенной модели подпрограмм в программировании.

Это вызывает удивление, но конечные автоматы, несмотря на доказанную функциональную эквивалентность той же модели граф-схем алгоритмов (ГСА) (подробнее об этом см. [4]), обходится без аналогичного понятия. Хотя, скорее всего, манипуляция часто огромным числом состояний – одно из последствий такого состояния дел с моделью.

Рассмотренные приемы компактного описания автоматов, как мы убедились, дают эффект. Тем не менее, понятно, что с практической точки зрения этого явно недостаточно. Модель, аналогичную модели подпрограмм, как доказавшую свою эффективность, желательно иметь и здесь. Пока же можно констатировать, что практики как-то выкручиваются, а теоретики – «корректно» избегают решения данной проблемы.

Тем не менее, попытки введения вложенной иерархии в автоматную модель были, но к желаемому результату они так и не привели. А предлагаемая вложенная иерархия для автоматов, например, в UML представляет всего лишь попытку решить «интуитивно», используя, возможно, опыт программистов, актуальную, но уже явно перезревшую проблему.

3.1. МОДЕЛЬ ВЛОЖЕННЫХ АВТОМАТОВ

Понятие *вызываемого* автомата является полным функциональным аналогом понятия подпрограммы. Модель автомата, включающая вызываемые автоматы, ценна не только своей практической направленностью. Так, рекурсивные алгоритмы без модели с вложенной иерархией не поддается прямой реализации, а известные методы преобразования рекурсивных алгоритмов к одноуровневому виду достаточно сложны и/или не столь очевидны, как того хотелось бы.

Определение 7. Вложенным автоматом будем называть двойку объектов $A = \langle M, m_0 \rangle$, где

M – конечное непустое множество дизъюнктивных автоматов,

m_0 – головной автомат (автомат верхнего уровня), $m_0 \in M$;

$m_0 = \langle S, X, Y, Z, s_0, \delta, \lambda, \alpha \rangle$, где

S – конечное непустое множество внутренних состояний;

X – конечное непустое множество входных сигналов;

Y – конечное непустое множество выходных сигналов;

Z – конечное непустое множество выходных сигналов автомата, порождающих вложенные автоматы, $Z \subseteq Y$;

$s_0 \in S_0$ – начальное состояние;

δ – функция переходов;

λ – функция выходов;

$\alpha: Z \rightarrow M$ – отображение выходных сигналов, порождающих **вызываемые автоматы** $m_i \in M$, $i = 1, 2, \dots, l$

Каждый вызываемый автомат $m_i \in M$, $i = 1, \dots, l$ определяется в свою очередь как:

$m_i = \langle S_i, X_i, Y_i, Z_i, s_0, s_{00}, \delta_i, \lambda_i, \alpha_i \rangle$, где

S_i – конечное непустое множество внутренних состояний;

X_i – конечное непустое множество входных сигналов, $X_i \subseteq X$;

Y_i – конечное непустое множество выходных сигналов, $Y_i \subseteq Y$;

Z_i – конечное непустое множество выходных сигналов автомата m_0 , порождающих вложенные автоматы, $Z_i \subseteq Y_i$;

$s_0 \in S_i$ – начальное состояние;

$s_{00} \in S_i$ – заключительное состояние;

$\alpha_i: Z_i \rightarrow M$ – отображение выходных сигналов, порождающих вызываемые автоматы $m_i \in M$, $i = 1, 2, \dots, l$. Реализацию элемента отображения α назовем **операцией вложения** вызываемого автомата, а номер уровня вложения $1, \dots, k$ – **рангом вложенности**.

Легко видеть, что если X, Y – множества соответственно входных и выходных сигналов (каналов) вложенного автомата A , включающего головной автомат и l вызываемых автоматов, то объединение всех входных/выходных сигналов автоматов

множества M представляет множества входных – X и выходных – Y сигналов автомата A , т.е.

$$(5) \quad X = \bigcup_{i=0}^l X_i, \bigcap_{i=0}^l X_i \neq \emptyset, \text{ где } X_i \in m_i, \quad i=1, \dots, l,$$

$$(6) \quad Y = \bigcup_{i=0}^l Y_i, \bigcap_{i=0}^l Y_i \neq \emptyset, \text{ где } Y_i \in m_i, \quad i=1, \dots, l.$$

Замечание 9. В общем случае пересечение сигналов автоматов $m_i \in M$ не пусто.

Операция вложения автомата $m_i \in M$ реализуется выдачей сигнала $z_i \in Z$, который формирует очередной уровень вложенной иерархии по отношению к текущему автомату. Так, аналогично принципу вложения подпрограмм, создается иерархия вложенных автоматов. Очередной вызываемый автомат реализует текущее функционирование автоматной модели. Его переход в заключительное состояние возвращает управлению автомату верхнего уровня, создавшего текущий вложенный автомат.

Отсюда следует необходимое, но совсем не обязательное, условие наличия заключительного состояния у вызываемого автомата. Просто при его отсутствии или если по каким-то причинам вызываемый автомат не перейдет в него, то, подобно закикливанию подпрограмм, не произойдет и возврата на предыдущий уровень иерархии модели.

Рассмотрим в деталях процедуру перехода с текущего уровня автоматной иерархии на уровень последующей вложенности модели. Во-первых, новыми текущими состояниями модели будут состояния вызываемого автомата. Во-вторых, [обычные] сигналы автомата верхнего уровня, выдаваемые на текущем переходе, выдаются при переходе в текущее состояние автомата, которое представлено начальным состоянием вызываемого автомата. В-третьих, сигнал, создающий вызываемый автомат может быть на переходе один и только один.

Замечание 10. Единственность на переходе сигнала, создающего вложенный автомат, определяется запретом нахождения автомата более чем в одном из своих внутренних состояний.

Текущим состоянием автомата, в которое должен перейти автомат после выхода из вложенного автомата, станет состояние автомата предыдущего уровня. Именно в него был направлен переход, создавший вызываемый автомат текущего уровня иерархии.

Пример 2. Пусть задан вложенный автомат $A = \langle M, m_0 \rangle$, где $M = \{m_0, m_1\}$, головной автомат $m_0 = (X_0, Q_0, Y_0, 1, F_0(x \in X_0 / y \in Y_0))$, где $X_0 = \{x1\}$, $Q_0 = \{0, 1\}$, $Y_0 = \{y1, z1\}$, вызываемый автомат $m_1 = (X_1, Q_1, Y_1, s5, 00, F_1(x \in X_1 / y \in Y_1))$, где $X_1 = \{x3\}$, $Q_1 = \{s5, 00\}$, $Y_1 = \{y1, z1\}$,

$\alpha_0: \{(z1, m1)\}$,

отображение головного автомата F_0 определяется следующим образом:

$1 = \{0(\wedge x1 / y1z1)\}$.

$0 = \{1(x1)\}$,

отображение вызываемого автомата F_1 следующее:

$s5 = \{s5(\wedge x3), 00(x3)\}$.

Графы головного автомата m_0 и вызываемого автомата m_1 показаны на рис.3. Одновременно рис.3 разъясняет функционирование вложенного автомата и графически демонстрирует алгоритм нахождения эквивалентного вложенному автомату обычного одноуровневого автомата. Легко видеть, что процедура нахождения эквивалентного автомата однозначна для любого вложенного автомата (вынесем пока рекурсивное вложение автоматов за рамки данной работы).

Теорема 4. *Любой вложенный ДНФ КА имеет единственный эквивалентный одноуровневый ДНФ КА.*

Доказывается однозначностью представленной на рис.3 процедуры последовательного приведения вложенного автомата к обычному автомату. Алгоритм подобного приведения состоит в последовательной «развертке» вызываемых автоматов, начиная с самого низкого уровня вложения – от k -го ранга вложенности к 1-му.

В отличие от предложенной модели, UML допускает, так называемые, ортогональные состояния, в которых ему разрешено пребывать одновременно в нескольких состояниях. Так

«формализуется» механизм параллельной работы автоматов «в контексте включающего объекта»[11].

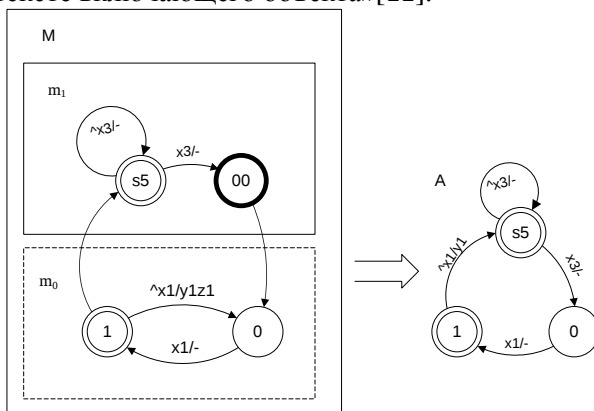


Рис.3. Пример вложенного автомата.

Замечание 11. Требование единственности операции вложения для дизъюнктивных автоматов отнюдь не запрещает создания на этом же переходе множества **параллельно** функционирующих автоматов. Но это тема, обсуждение которой выходит пока за рамки рассматриваемых моделей автоматов.

Замечание 12. Операции вложения автоматов разрешает рекурсию, т.е. вложение в автомат самого себя – автомата текущего уровня иерархии.

Однако рекурсивная модель требует передачи контекста данных на следующий уровень иерархии и доступа к нему же при завершении работы текущего уровня вложения. Все это не так уж сложно сделать (см. [9]), но подобно теме параллелизма автоматов выходит за рамки вопросов, обсуждаемых в данной работе.

3.2.МОДЕЛЬ ИНЕРЦИОННЫХ АВТОМАТОВ

Инерционный автомат представляет модель, которая «помнит» текущее состояние, чтобы иметь возможность при необходимости в него вернуться. Подобный вариант развития событий в рамках UML реализуют с помощью *исторических состояний*. Но, как справедливо отмечено в [13], модели подобного вида

вполне можно реализовать обычными автоматами, но ... «это будет некрасиво» (см. [13], стр.327).

Нас же заботит не столько «красота», сколько ситуации, которые без *инерционных* автоматов сложно разрешить без существенного усложнения алгоритмов.

*Определение 8. Назовем **инерционным автоматом** модель вложенного автомата $A = \langle M, t_0 \rangle$, у которого среди вызываемых автоматов есть хотя бы один $t_i \in M$ с дополнительным **заключительным состоянием инерционного типа**, определяемый как:*

$t_i = \langle S_i, X_i, Y_i, Z_i, s_0, s_{00}, s_{xx}, \delta_i, \lambda_i, \alpha_i \rangle$, где

S_i – конечное непустое множество внутренних состояний;

X_i – конечное непустое множество входных сигналов, $X_i \subseteq X$;

Y_i – конечное непустое множество выходных сигналов, $Y_i \subseteq Y$;

Z_i – конечное непустое множество выходных сигналов автомата t_i , порождающих вложенные автоматы, $Z_i \subseteq Y_i$;

s_0 – начальное состояние;

s_{00} – заключительное состояние обычного типа;

s_{xx} – заключительное состояние инерционного типа;

$\alpha_i: Z_i \rightarrow M$ – отображение выходных сигналов, порождающих вызываемые автоматы $t_i \in M$, $i = 1, 2, \dots, l$.

Инерционный автомат – это вложенный автомат, но включающий хотя бы один вызываемый автомат, называемый *инерционным вызываемым автоматом*. Последний, наряду с обычным заключительным состоянием, содержит заключительное состояние, переход в которое определяет иную процедуру возврата, а именно, – возврат в состояние, в котором был создан вызываемый автомат. Так реализуется фактически отмена перехода в следующее состояние (что, подчеркнем, невозможно по определению в «классике»).

Пример 3 представляет «схему» работы инерционного вложенного автомата. Далее будет рассмотрен еще один пример инерционного автомата – автоматной модели элемента задержки инерционного вида (послужившей, кстати, прототипом для инерционной автоматной модели), демонстрирующий проблему уже в деталях.

Пример 3. Пусть задан вложенный инерционный автомат $A = \langle M, m_0 \rangle$, где $M = \{m_0, m_1\}$, головной автомат $m_0 = (X_0, Q_0, Y_0, 1, F_0(x \in X_0/y \in Y_0))$, где $X_0 = \{x1\}$, $Q_0 = \{0, 1\}$, $Y_0 = \{y1, z1\}$, вызываемый инерционный автомат $m_1 = (X_1, Q_1, Y_1, s5, 00, XX, F_1(x \in X_1/y \in Y_1))$, где $X_1 = \{x1, x3\}$, $Q_1 = \{s5, 00, XX\}$, $Y_1 = \{-\}$,

$\alpha_0: \{(z1, m1)\}$,

отображение головного автомата F_0 определяется следующим образом:

$1 = \{0(\wedge x1/y1)\}$.

$0 = \{1(x1/z1)\}$,

отображение вызываемого инерционного автомата F_1 следующее:

$s5 = \{s5(x1 \wedge x3), 00(x3), XX(\wedge x1 \wedge x3)\}$.

На рис.4 показаны графы вложенного инерционного автомата. Видно, что автомат $m1$ имеет два заключительных состояния. Одно из них – с именем «00» соответствует обычному заключительному состоянию, а с именем «XX» - инерционному. Чтобы на графе отличить инерционный переход от обычного перехода, будем его изображать пунктирной дугой. Такое изображение будет предупреждать о том, что это переход, который *не обязательно* (!) приводит к смене текущего состояния.

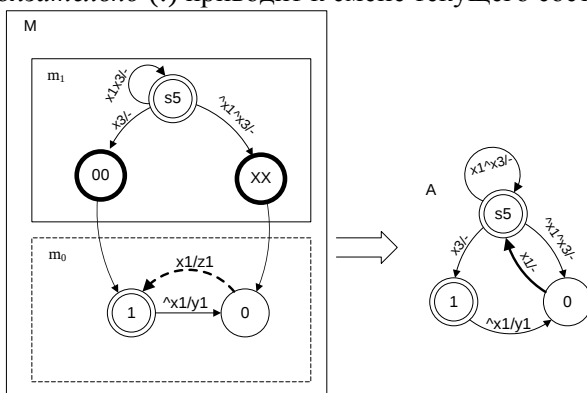


Рис.4. Пример инерционного автомата.

Обратим еще раз внимание на то, что у классического автомата срабатывание обычного перехода *обязательно* (!) ведет к

смене текущего состояния. У инерционного же автомата есть переходы, которые на первый взгляд нарушают подобное соглашение о работе автоматов.

Собственно в этом – возможности возврата в текущее состояние и состоит принципиальное отличие инерционных автоматов от автоматов обычного вида. Но, подчеркнем, что существенным является также то, что инерционные автоматы рассматриваются только в контексте вложенных автоматов.

Графы на рис.4 представляют наглядную процедуру приведения инерционного автомата к эквивалентному одноуровневному автомату. С этой целью необходимо:

1) направить дугу, вызывающую вложенный автомат, в его начальное состояние;

2) его обычный завершающий переход (на рис.4 это переход с именем «00») совместить с состоянием назначения автомата верхнего уровня;

3) инерционное заключительное состояние (на рис.4 это состояние с именем «XX») совместить с состоянием, породившим вложенный [инерционный] автомат.

Теорема 5. Любой вложенный инерционный ДНФ КА имеет единственный эквивалентный одноуровневый ДНФ КА.

Доказывается однозначностью представленной на рис.4 процедуры последовательного приведения инерционного вложенного автомата к обычному автомату. Алгоритм подобного приведения состоит в «развертке» обычных и инерционных вызываемых автоматов, начиная с самого низкого уровня вложения – от k -го ранга вложенности к 1-му.

Далее рассмотрим пример модели дискретного элемента, который отражает актуальность модели инерционного типа. Ситуацию, когда без нее сложно создать модель, которая выглядит не просто «красиво», а весьма просто и ясно (а в этом, скорее всего, и есть ее «красота»!).

Именно инерционный автомат в подобной ситуации дает необходимую ясность и адекватность функционирования и определяет надежность работы подобных дискретных моделей.

4. Конечно-автоматная модель задержки

Пусть стоит задача создания модели логического элемента, передающего без преобразования значение входного сигнала на выход. Параметры сигналов, определяющие при этом значения задержек переключения элемента, показаны на временной диаграмме рис.5.

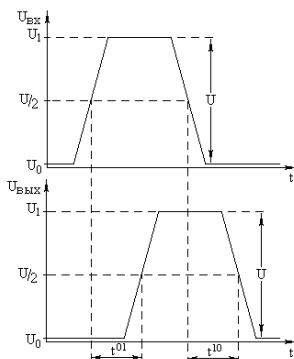


Рис. 5. Определение времени задержки логического элемента

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.6.

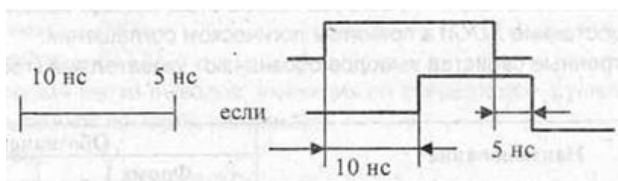


Рис. 6. Изображение задержки по ГОСТ 2.743-91

4.1.МОДЕЛЬ В ФОРМЕ ДНФ КА

Создавая модель элемента задержки в форме конечного автомата, удобно моделировать значение величины задержек числом дискретных тактов модели автомата *синхронного* типа (определение синхронных автоматов см. в [5]).

Самая простая автоматная модель *единичной задержки*, времени переключений которой определяются единичным дискретным тактом синхронной модели, показана на рис.7 (см. для сравнения автоматную модель задержки, приведенную в [8]).

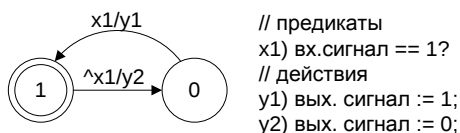


Рис. 7. Модель единичной задержки в форме автомата Мили

Рассматривая далее автоматные модели, будем иметь в виду, прежде всего, их *программную* реализацию, допуская также и *аппаратную* реализацию моделей [4, 10]. При этом будем опускать описание методов и свойств, например, объектно-ориентированной реализации модели (см. [9]), рассматривая в целях простоты поведение объектов, ассоциируемое с автоматной моделью управления. В практике программирования реализация подобных элементов (методов и свойств) объектной модели не вызывает каких-либо проблем и в рамках дальнейшего обсуждения не требует подробных пояснений.

На рис.8 показаны две модели, так называемой, *транспортной задержки* [3] с моделированием значений t^{01} и t^{10} . Одна в форме автомата Мили (рис.8а), вторая – в форме совмещенной модели Мили-Мура (рис. 8б). Транспортная задержка в отличие от *инерционной задержки*, передающей на выход только изменение сигналов длительностью не менее длительности задержек, передает на выход любое изменение сигнала на входе[3, 12].

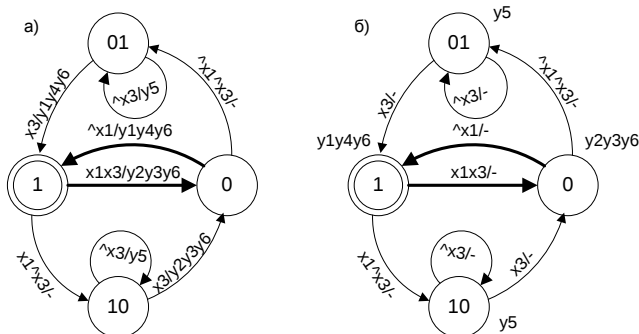


Рис. 8. Модели элемента с транспортными задержками в форме автоматов Мили (а) и Мили-Мура (б)

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 параметров и возвращаемого значения). Сигналы y_1 , y_2 присваивают значению свойству – переменной, поставленной в соответствие выходу элемента, y_5 – увеличивает счетчик тактов, который сбрасывается сигналом y_6 .

Модели на рис.8 отражают отличия в функционировании в целом эквивалентных моделей – автоматов Мили и Мура (напомним, что совмещенная модель Мили-Мура объединяет в себе функционирование обоих типов моделей). Считается, что модель Мили гибче и в силу этого может содержать меньшее число состояний. При этом автомат Мура 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 невольно 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 у5. Дополнительно модель упростилась за счет исключения повторов записи сигналов на переходах модели Мили.

Подсчет тактов в состояниях, моделирующих задержки, усложнил формальную модель (сравним модель без задержек на рис.7 с моделью с задержками на рис.8). Неизбежное для подобной [классической] модели автомата введение дополнительных внутренних состояний привело к утрате удобного свойства – ассоциации состояний модели с состоянием выхода элемента. Это стало следствием **обязательного** введения в модель локальной переменной и сигналов, устанавливающих ее значение. В отличие от этого для модели на рис.7 сигналы у1 и у2 лишь отражали подобную возможность, но не были необходимостью.

4.1.ВЛОЖЕННАЯ МОДЕЛЬ ЗАДЕРЖКИ

Используя механизм вложения моделей, аналогичный механизму обычных подпрограмм, можно привести довольно сложную модель, представленную на рис.8, фактически к ее простейшему исходному виду на рис.7.

Модель конечного автомата, использующая вложение автоматов, показана на рис.9. Здесь вызываемый на переходе автомат реализует задержку, подсчитывая дискретные такты, возвращая (при равенстве счетчика тактов значению задержки) управление модели верхнего уровня.

С использованием механизма вложения автоматов модель на верхнем уровне иерархии рис.9а по структуре совпадает с моделью на рис.7. При этом моделирующий задержки вложенный автомат на рис.9б представляет собой гораздо более простой автомат по сравнению с одноуровневой моделью на рис.8. Более того, уже как «библиотечный автомат», он может быть использован в любой другой автоматной модели.

Замечание 13. В целях упрощения графического представления автоматной модели далее будет использован прием – склейка, представляющий замену нескольких дуг переходов, принадлежащих общим состояниям и выдающих одинаковые выходные сигналы, одной дугой.

На рис.9а этому соответствуют две дуги, показанные пунктирными линиями, которые заменены одной дугой, изображенной утолщенной линией (этому соответствует замена переходов $x1x3/-$ и $x1\wedge x3/z1$ переходом $x1/z1$). При программной реализации подобная замена реализуется в действии, создающем вложенный автомат – $z1$. В результате форма автомата фактически один в один соответствует его графической форме описания.

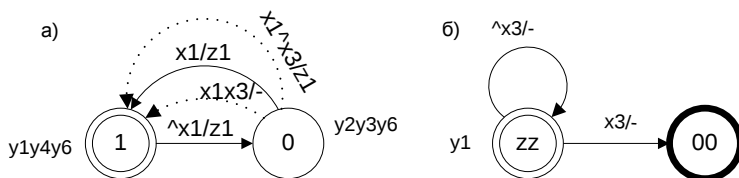


Рис. 9. Модель элемента(а) с транспортными задержками (б) в форме вложенных автоматов

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 автомат переходит в состояние «00», возвращая тем самым управление предыдущему уровню иерархии.

4.2.ИНЕРЦИОННАЯ МОДЕЛЬ ЗАДЕРЖКИ

Логический элемент, реализующий принцип инерционной задержки, более адекватен реальным объектам, которые характеризуются *полосой пропускания*, не позволяющей им реагировать на слишком быстрые изменения входных сигналов[5]. Модель элемента инерционной задержки в форме совмещенной автоматной модели Мили-Мура показана на рис.10.

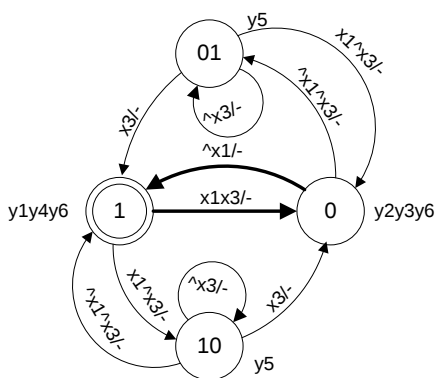


Рис. 10. Автоматная модель элемента с инерционными задержками

Из сравнения данной модели с моделью на рис.8б очевидно их 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.

Но даже такое, казалось бы, незначительное отличие оказывает качественное влияние и не позволяет использовать механизм обычного вызова автоматов.

Замечание 14. В существующей алгоритмической модели после выхода из подпрограммы вычисления возобновляются со следующей после вызова подпрограммы команды/инструкции, но, заметим, никак не в точку вызова подпрограммы.

Для решения возникшей проблемы предлагается использовать модель, которая позволяет вернуть управление в точку, предшествующую вызову подпрограммы, т.е. модель инерционных автоматов. Для чего введем в модель обычного вызываемого автомата еще одно завершающее состояние с именем «XX». Модель, включающая подобные вложенные автоматы (с двумя видами заключительных состояний), показана на рис.11.

Теперь на верхнем уровне представления лишь наличие инерционных переходов отличает модель на рис.11а от модели на рис.9а. 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».

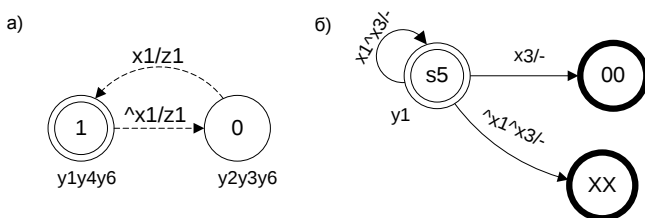


Рис. 11. Модель элемента (а) с инерционными задержками (б) в форме вложенных автоматов

Подчеркнем, что использование автомата инерционного вида определяется трактовкой функционирования автоматной модели: инерционный автомат имеет переходы, запуск которых определяет процедуру *отложенного перехода* в новое состояние.

Подобных переходов у обычного автомата нет: сработавший переход **обязательно** (!) приводит к смене текущего состояния (в случае петли оно просто совпадает с текущим состоянием).

Легко видеть, что любая из вложенных моделей путем обратных преобразований приводится к эквивалентному обычному автомату. Однозначность подобных преобразований моделей при приведении их к классической модели, доказывает, что функционирование автоматов, имеющих вложенные автоматы рассмотренных выше видов, эквивалентно функционированию соответствующих одноуровневых конечных автоматов.

5. Заключение

Упрощение не должно вести к потере качества – способности точно и адекватно представлять дискретную модель объекта и/или вычислительный алгоритм. Сформулированные в рамках

работы определения моделей конечных автоматов в дизъюнктивной форме служат именно этим целям.

Научная ценность введенных дизъюнктивных моделей конечного автомата – в формировании модели, порождающей и новую модель вложения автоматов, и новую модель инерционного конечного автомата. Последний фактически определяет соответствующий класс инерционных алгоритмов.

Практическая ценность описанных моделей состоит в формировании технологии проектирования, которая позволяет наглядно, адекватно, эффективно и, что немаловажно, в компактной форме решать проблемы описания и проектирования дискретных систем. При этом наследуются в полной мере свойства базовой модели, что позволяет применять без ограничений на практике ее теорию.

Реализацию технологии проектирования программных систем, базирующейся на дизъюнктивной автоматной модели управления и описанных выше моделей, представляет визуальная среда проектирования программ – ВКПа[9]. Результаты тестирования в ней описанных моделей задержек показаны на рис.12. Видно, что они отражают свойство реального элемента устранять дребезг слишком быстрого изменения входного сигнала.

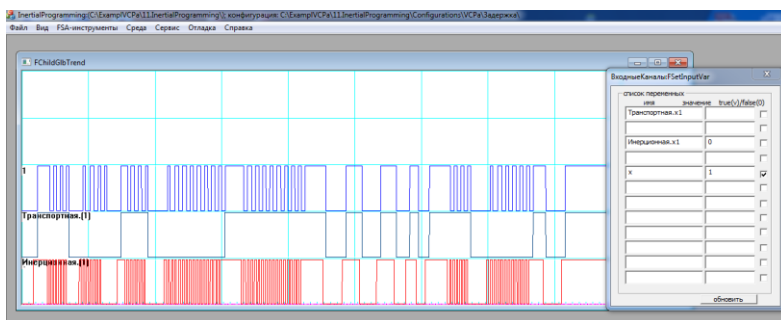


Рис. 12. Тестирование моделей задержек в среде ВКПа.

Как мы убедились, инерционная модель задержки в рамках соответствующей модели на верхнем уровне описания, реализуя в целом достаточно сложный алгоритм обработки «инерцион-

ных сигналов», фактически не отличается по сложности от самого простого варианта модели (ср. модель единичной задержки на рис.7 с головным автоматом инерционной задержки на рис.11а).

По мнению автора, направление дальнейших исследований и экспериментов состоит в формализации алгоритмической модели параллельных вычислений, в которой модель дизъюнктивного автомата используется в качестве модели управления, а сеть дизъюнктивных автоматов – в качестве модели управления для множества параллельных процессов.

Литература

1. АМБАРЦУМЯН А.А., ЗАПОЛЬСКИХ Е.Н. *Об одном подходе к временной декомпозиции автоматов. I*, Автомат. и телемех., 1981, выпуск 2, 135-144
2. АМБАРЦУМЯН А.А., ЗАПОЛЬСКИХ Е.Н. *Об одном подходе к временной декомпозиции автоматов. II*, Автомат. и телемех., 1981, выпуск 3, 112-121
3. АРМСТРОНГ ДЖ.Р. *Моделирование цифровых систем на языке VHDL*: Пер с англ./М.: Мир, 1992. – 175 с.
4. БАРАНОВ С.И. *Синтез микропрограммных автоматов*. – Л.: Энергия, 1979. – 232с.
5. ГЛУШКОВ В.М. *Введение в кибернетику*. Изд-во АН Украинской ССР. К.: 1964. - 324с.
6. ЗАКРЕВСКИЙ А. Д., ПОТТОСИН Ю. В., ЧЕРЕМИСИНОВА Л. Д. *Логические основы проектирования дискретных устройств*. — М.: ФИЗМАТЛИТ, 2007. - 592 с.
7. КАРПОВ Ю.Г. *Теория автоматов*. – СПб.: Питер, 2003. – 208 с.
8. КУДРЯВЦЕВ В.Б., АЛЕШИН С.В., ПОДКОЛЗИН А.С. *Введение в теорию автоматов* – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 320 с.
9. 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.

10. МАЙОРОВ С.А., НОВИКОВ Г.И. *Принципы организации цифровых машин*. – Л.: Машиностроение, 1974. – 431 с.
11. МЕЛИХОВ А.Н. *Ориентированные графы и конечные автоматы*. – М.: Наука, 1971. – 416 с.
12. МИЛЛЕР Р. *Теория переключательных схем, т.2 – Последовательностные схемы и машины* : Пер. с англ. – М.: Наука, 1971. – 304с.
13. РАМБО ДЖ., ЯКОБСОН А., БУЧ Г. *Язык UML. Руководство пользователя. Второе издание*. Академия АЙТИ: Москва, 2007. – 493 с.
14. ЯБЛОНСКИЙ С.В. *Введение в дискретную математику*: Учеб. пособие для вузов. – 2-е изд., перераб. и доп. – М.: Наука. Гл. ред. физ.-мат. Лит. – 384 с.

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 the description of model of disjunctive automatic machines and created on their base of model of the enclosed automatic machines which are a basis of new model of the inertial final automatic machine is given. Equivalence of the entered models to the classical final automatic machine is proved. On examples properties of the entered models, actual algorithmizations for practice of discrete models of systems are considered.

Keywords: the automatic machine with an abstract condition, the disjunctive automatic machine, the enclosed automatic machine, the inertial automatic machine, model of an element of a delay.

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

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