УДК 519.713
ББК 32.815

ДИЗЪЮНКТИВНАЯ ФОРМА СТРУКТУРНЫХ АВТОМАТОВ

Любченко В.С. [1]
(*Учреждение Российской академии наук
Институт проблем управления РАН, Москва*)

Рассмотрена абстрактная модель вычислений в автоматной форме и сформулирована дизъюнктивная форма представления структурных автоматов. Предложена сетевая модель и алгебраическая операция композиции на базе структурных автоматов. Описана автоматная модель управления параллельных вычислений. Приводится сравнение с сетями Петри.

Ключевые слова: теория конечных автоматов, дизъюнктивная нормальная форма структурных автоматов, операция композиции автоматов, параллельная модель вычислений.

1. Введение

У моделей дискретных систем существуют проблемы, связанные с проектированием сложных многокомпонентных решений. Не являются исключением в этом смысле и модели, представленные конечными автоматами. Для них ситуация усугубляется предубеждением об их ограниченности для реализации вычислительных алгоритмов вообще и систем параллельного типа в частности. Но есть множество причин не ослаблять внимание к модели, которую часто необоснованно отвергают в ситуациях, с которыми лучше ее, скорее всего, ни кто и не справится.

Необходимо признать, что прямо или косвенно проблемы автоматных моделей приводят в конечном итоге к выбору дру-

---

[1] *Любченко Вячеслав Селиверстович, ведущий инженер* (sllubch@*mail.ru*).

гих моделей. Для примера можно привести обоснование преимущества сетей Петри перед автоматами, приведенные в книге Дж. Питерсона (см. п.3.1.1 в [10]).

Расширение возможностей автоматов для описания и реализации, как отдельных вычислительных процессов, так и их множества, представляет не только теоретическое, но большое практическое значение.

2. О проблемах теории автоматов

Общая теория автоматов подразделяется на теорию абстрактных автоматов и теорию структурных автоматов. Характерным является разбиение автоматных подходов на так называемые группы микро- и макроподходов. Классический подход теории автоматов предусматривает двухэтапное проектирование, когда на первом этапе строится абстрактная модель решения в виде преобразователя или акцептора, а на втором рассматривается ее реализация в форме структурной модели (см. подробнее в [4, 8]).

Типовые задачи структурной теории автоматов рассматривают вопросы кодирования входной, выходной информации, внутренних состояний автоматной модели и синтеза схемы автомата из выбранного набора элементарных автоматов для созданной перед этим абстрактной модели. В результате может сложиться мнение о подчиненной роли структурных автоматов по отношению к абстрактным.

На самом деле теория структурных автоматов явно шире, чем это может показаться на первый взгляд. В ней решаются вопросы, которые не имеют решения в теории абстрактных автоматов (подробнее об этом см. [2], стр.166), но имеют большое значение для проектирования сложных многокомпонентных систем. В первую очередь к ним относится рассмотрение различных композиций автоматов.

С точки зрения сложных систем структурные автоматы имеют больше возможностей для отражения структуры и описания взаимодействия компонент. Однако, если при классическом подходе на этапе структурного синтеза нам уже известно поведение системы, представленной абстрактным автоматом, то здесь ситуация обычно обратная – известны сами компоненты (но часто, заметим, не их модели) и схема, определяющая их взаимодействие. Основной проблемой при этом становится не выбор компонент (из функционально полного набора элементарных автоматов) и синтез схемы, а создание моделей компонент, если они не известны, а затем анализ и определение поведение системы в целом.

Создать автоматную модель сложной системы и решать задачи ее анализа и синтеза станет много легче, если структурный автомат рассматривать как самостоятельную абстрактную модель, снимающую ограничения на число входных/выходных каналов. При этом вопросы кодирования состояний, синтез схемы структурного автомата, реализующего некий абстрактный автомат, выносятся за рамки подобной «абстрактной структурной модели». Но при этом результаты, полученные при решении проблем композиций структурных автоматов, можно распространить уже на макроуровень проектирования систем. С точки зрения теории структурных автоматов это соответствует рассмотрению схем, построенных теперь уже часто далеко не из элементарных «абстрактных структурных автоматов».

Казалось бы, простое изменение подхода к проектированию не должно привести к проблемам создания структурной модели (схемы) сложной системы и анализа ее поведения. Тем более, что можно использовать уже проверенную практикой теорию. Но, к сожалению, это не так, т.к. нужно получить ответы на вопросы, которые в силу определенных причин были упущены, упрощены или попросту проигнорированы при решении аналогичных задач теории структурных автоматов.

3. Пути решения проблем автоматных моделей

Рассматривая истоки проблем автоматных моделей в их приложении к сложным системам, в первую очередь, видимо, необходимо решить проблему элементов «мгновенного» действия (стр.11 в [4]). В их роли обычно выступают элементарные автоматы с одним состоянием. В то же время мы не наблюдаем ни мгновенных действий, ни мгновенных перемещений реальных объектов (или, как минимум, они пока не известны). Поэтому на уровне автоматной модели, скорее, можно и нужно говорить о разном дискретном времени отдельных компонент, чем допускать их «мгновенную» работу.

Другая актуальная проблема - циклические цепи из автоматов. Сейчас она решается включением «в разрыв» подобной цепи автомата Мура. Ими могут быть, например, задержки. Но это явно навязанная процедура. В целях соответствия структурной модели ее реальному аналогу ее желательно исключить. В теории же подобная процедура представляет одно из условий корректного построения схем классических структурных автоматов. И, что примечательно, сделать это не так уж и сложно.

Безусловно, сложные вычислительные задачи нуждаются в универсальной вычислительной модели, сформулированной в автоматной форме для отдельной компоненты. Она должна реализовать вычислительный процесс любой сложности, а не только представленный логическими схемами. В подобную «автоматную машину» необходимо включить модели памяти и операторов, связав их с моделью управления в форме структурного автомата.

Необходимо четко определить место автоматной модели в пределах абстрактной алгоритмической модели вычислений. Модель блока управления – это та часть «машины», которая наиболее адекватно формализуется моделью конечного автомата. При этом вычислительная нагрузка, т.е. выполнение процедур вычисления и операций с данными, возлагается на операторы и память – две другие компоненты, составляющие в сумме с управлением общую модель абстрактных вычислений.

Необходимо также уточнение формальной модели управления сложной системы, представленной не только одним автоматом, но и в общем случае – схемой (сетью) из автоматов. Заметим, что на уровне вычислений это может быть сеть «машин».

С точки зрения формальных преобразований дискретное время схемы должно быть единым. Существенно, что подобное соглашение не ограничивает общность модели (подробнее см. стр.171 в [2]). Тем не менее, разные схемы могут иметь свое время. Вполне допускается различие дискретного времени даже для компонентов схемы. Другое дело, что в этих случаях будут определенные проблемы, например, связанные с анализом поведения модели (нахождение эквивалентного автомата для сети).

Также невозможно обойтись без пересмотра момента выдачи выходных сигналов. Исходя из определения автоматной модели, они выдаются или «мгновенно», т.е. в самом начале дискретного такта – для автоматов Мили, или в конце такта после перехода в новое состояние – для автоматов Мура.

Многие проблемы снимаются, если алгоритм работы структурных автоматов сети (схемы) в пределах текущего дискретного такта будет следующим: сначала все автоматы схемы выполняют анализ входных каналов и только после этого они переходят в следующее состояние и выдают выходные сигналы.

Подобный сдвиг момента выдачи сигналов относительно начала дискретного такта решает проблему циклических связей/петель или «мгновенной зависимости» сигналов (см. [4]). В результате снимается довольно серьезное ограничение, связанное с обязательным включением в петлю из автоматов хотя бы одного автомата Мура (см. подробнее в [2]).

Подобная трактовка закона функционирования автоматов, как отдельного автомата, так и схем в целом, не влияет на теоретические результаты, уже полученные в рамках теории конечных автоматов. Но она имеет решающее значение с точки зрения практической реализации модели дискретной системы, ее адекватности реальным объектам.

Обоснованность предлагаемых расширений [структурной] автоматной модели подтверждена результатами практической ее реализации в рамках программной технологии проектирования сложных программных систем параллельного типа, названной технологией автоматного Визульно-Компонентного программирования (сокращенно – ВКПа)[7].

Решив упомянутые выше проблемы, в рамках структурной модели можно создать компактную, ясную, структурно и функционально адекватную реальным системам алгоритмическую модель и расширить теорию параллельных систем новыми возможностями их анализа и синтеза.

4. Абстрактная машина Мили

Ограниченные вычислительные возможности автоматной модели легко снимаются подключением к автомату в той или иной форме памяти. Например, если ею будет бесконечная лента, то получится машина Тьюринга[4]. Аналогичное решение – автоматы над внутренней памятью. Но больший интерес представляет формализм, представленный абстрактной моделью вычислителя в форме двух взаимодействующих автоматов - управляющего и операционного[3].

Обычно управляющий автомат абстрактного вычислителя представлен автоматом Мили, а в качестве операционного автомата выбирают автомат Мура. Операционный автомат включает память, и множество блоков, реализующих операции/микрооперации, отождествленные с выходными сигналами управляющего автомата – y_j, $j = 1, \ldots, m$; m – число выходных каналов управляющего автомата. При этом входными сигналами - x_i $i = 1, \ldots, n$, n – число входных каналов управляющего автомата являются значения различных логических условий, формируемых операционным автоматом.

Расширим понятие микроопераций и логических условий до [программных] функций, отождествленных каналам управляющего автомата. Функции, отождествленные выходным каналам, назовем действиями – $y_j()$, входным каналам – предикатами – $x_i()$.

Действия могут реализовывать достаточно сложные по сравнению с микрооперациями программные процедуры. В свою очередь они могут быть представлены моделью автоматных вычислений. Так появляется понятие вложенного автомата.

Функциональное назначение предикатов выполнить анализ заданного условия и вернуть результат в булевской форме. Это будет значение «истина», если условие выполнено, и «ложь» в противном случае. Понятно, что исполнение предикатов и действий, кроме действий, представленных автоматной моделью, должно быть завершено до истечения времени текущего дискретного такта.

Назовем представленную выше абстрактную модель вычислителя по имени базовой модели управляющего автомата, т.е. **машиной Мили**. Пример программы, эквивалентной машине Тьюринга из [4], на С-подобном языке, реализуемой машиной Мили, представлен на листинге. 1.

```
// П р и м е р 10 (стр.30, [4]).
// ОПЕРАЦИОННЫЙ АВТОМАТ
char  chType[] = {0,0,0,0,1,1,1,1,1,1,0,0,0}; // конфигурация ленты
int nInd = 4;                                  // текущий «адрес» ленты
// предикаты
x1) return chType[nInd] == “0”;// символ равен «0»?
x2) return chType[nInd] == “1”;             // символ равен «1»?
// действия
y1) nInd++;                    // R – сдвиг «головки» вправо
y2) nInd--;                    // L – сдвиг «головки» влево
y3) chType[nInd] = 0;          // присвоить ячейке ленты «0»
y4) chType[nInd] = 1;          // присвоить ячейке ленты «1»
y5) chType[nInd] = 1; nInd++;  // изменить, переместиться вправо
// УПРАВЛЯЮЩИЙ АВТОМАТ
// таблица переходов автомата Мили
// формат: тек.сост., след.сост., усл.перехода, действия
g0, g2, x1, y3,
g0, g1, x2, y4,
g1, g2, x1, y4,
g1, g0, x2, y4y1,
g2, g2, x1, y3,
g2, g0, x2, y4;
```

Листинг. 1. Пример программы для машины Мили

Приведенный пример выявляет проблемы, которые необходимо решить при реализации машины Мили. В первую очередь они связаны с функциями, отождествленными с каналами автомата. Исходя из уже сказанного понятно, что, во-первых, предикаты должны выполниться до запуска действий, а те и другие должны быть завершены в рамках дискретного такта.

Во-вторых, появляются проблемы параллельного исполнения операторов. И если с предикатами проблем нет, т.к. они не изменяют значение памяти, то исполнение действий, у которых имеются общие переменные, приведет к конфликту одновременного доступа.

Из примера легко видеть, что конфликты доступа к общим переменным не обязательно связаны с общими переменными. На примере перехода «g1, g0, x2, y4y1» видно, что действие y4(), связанное с выходным сигналом y4, изменяет значение элемента памяти «ленты», а действие y1(), связанное с сигналом y1, реализует «перемещение».

Исходя из логики работы машины, необходимо сначала изменить содержимое памяти, а потом переместиться на другую ячейку памяти (ленты). И, казалось бы, сама запись выходных сигналов перехода в виде «y4y1» говорит о том, что необходимо сначала изменить значение памяти, а только потом выполнить «сдвиг головки вправо» (увеличение индекса массива, моделирующего ленту).

Но сигналы по каналам управляющего автомата поступают и/или выдаются одновременно и потому соответственно одновременно, т.е. параллельно, должны быть запущены и отождествленные им функции. Если это соблюдено, то записи вида «y4y1» и «y1y4» эквивалентны. Кто из них «победит», добиваясь доступа к общей переменной, предсказать невозможно, да и, рассматривая формализм модели, не нужно.

Самое надежное – разнести конфликтующие действия по разным переходам автомата, указав явно последовательность их запуска. Другое решение – реализовать обобщенное действие, в котором явно определить последовательность изменения переменной и «перемещения по ленте» (см. действие y5(),листинг 1).

Есть также и другие не столь очевидные, но весьма важные с точки их разрешения проблемы, которые будут рассмотрены при более детальном взгляде на модель управляющего автомата, реализующего управление параллельными вычислениями в наиболее общем виде – в форме взаимодействующих параллельных процессов.

5. Дизъюнктивная форма конечных автоматов

Прежде чем перейти к рассмотрению модели сложной многокомпонентной системы, рассмотрим модель абстрактного структурного автомата (далее, если это ясно из контекста, просто структурного автомата), позволяющего представить автоматную модель компонента в более компактном виде, чем это позволяют обычные абстрактные автоматы.

В отличие от абстрактных автоматов, у которых множества X и Y являются множествами соответственно входных и выходных символов, в структурных автоматах они представлены *структурными алфавитами*, состоящими из элементарных конъюнкций логических переменных, соответствующих входным и выходным сигналам/каналам автоматов.

Рассматривая структурные автоматы, будем допускать наличие у них «пустых» наборов входных и выходных логических сигналов (переменных). Они будут представлены в структурных алфавитах символом «прочерк». Переход, помеченный «пустым» входным набором, представляет безусловный переход в следующее состояние. Такой переход у состояния может быть один и только один. Использование прочерка на месте выходного набора будет означать отсутствие выходных сигналов.

Введем по аналогии с нормальными формами булевых функций нормальные формы описания вполне (иногда говорят – полностью) определенных структурных автоматов. Это позволяет выделить содержательную часть информации о функционировании автомата, скрыв несущественную. Под существенной информацией понимаются переходы, изменяющие состояние автомата, и/или переходы с выдачей выходных сигналов пусть даже без перехода в новое состояние.

Напомним, что в аналогичной ситуации для булевых функций выделяют единичные значения функции, отбрасывая нулевые, для дизъюнктивных форм, и нулевые значения функции для конъюнктивных.

Назовем *дизъюнктивной нормальной формой* структурных конечных автоматов (ДНФ СКА) вполне определенные структурные автоматы Мили, переходы которых помечены элементарными конъюнкциями логических переменных, соответствующих входным/выходным сигналам автомата.

Модель ДНФ СКА содержит только *результативные* переходы, т.е. переходы, помеченные выходными сигналами, или переходы с прочерком на месте выходных сигналов, но изменяющие текущее состояние автомата.

Переходы, не включенные в нормальную форму автомата, представляют дополнение ДНФ СКА до вполне определенного структурного автомата. Формально подобное дополнение представляет автомат, состоящий из [изолированных] состояний, содержащих переходы в виде петель с прочерком на месте выходных сигналов.

На смысловом уровне модель ДНФ СКА отделяет явным образом действия автомата от ситуаций, когда изменение информации на входах автомата не ведет к какому-либо его изменению, связанному со сменой внутреннего состояния автомата и/или изменению состояния его выходов.

Автоматы ДНФ СКА, входные переходы которых помечены *конституентами* единиц (элементарные конъюнкции ранга m для автомата, имеющего m входных каналов), назовем *канонической* или *совершенной* формой представления дизъюнктивных нормальных форм структурных автоматов – ДКФ СКА (СДНФ СКА).

Любой вполне определенный дизъюнктивный структурный автомат M можно представить как объединение двух частичных автоматов: частичного автомата S в форме ДНФ СКА и частичного автомата ^S, состоящего из подмножества состояний автомата M с переходами в форме петель, где

(1) $M = S \cup \overline{S}$.

Ясно, что в случае, когда все переходы автомата результативные, M = S.

Покажем на примерах способы задания структурных автоматов в форме ДНФ СКА.

*Пример***.** Пусть задан вполне определенный структурный автомат A = (X, Q, Y, $s_0 \in Q$, $F(\tilde{x} \in X/\tilde{y} \in Y)$)>, где $X = \{\bar{x}_1\bar{x}_2, \bar{x}_1 x_2, x_1\bar{x}_2, x_1 x_2\}$, $Q = \{s0, s1\}$, $Y = \{y_1, y_2\}$, у которого отображение F множества Q в себя в аналитической форме (подробнее о формах представления автоматов см. в [9]) определяется следующим образом:

(2) F:

$$s_0 = \{s_1(\bar{x}_1\bar{x}_2/y_1)\}, \{s_1(\bar{x}_1x_2/y_1)\}, \{s_1(x_1\bar{x}_2/y_1)\}, \{s_0(x_1x_2/y_1)\},$$

$$s_1 = \{s_1(\bar{x}_1\bar{x}_2/y_1)\}, \{s_1(\bar{x}_1x_2/y_1)\}, \{s_1(x_1\bar{x}_2/y_1)\}, \{s_0(x_1x_2/y_1)\}.$$

Далее будем использовать более простую форму текстового описания автоматных отображений (далее просто отображений), заменяя символ «отрицания» символом ^. В этом случае элементарная конъюнкция $\bar{x}_1\bar{x}_2$ будет выглядеть как ^x1^x2. С учетом этого отображение (2) примет следующий вид:

(3) F:
s0 = {s1(^x1^x2/y1), s1(^x1x2/y1), s1(x1^x2/y1), s0(x1x2/y2)},
s1 = {s1(^x1^x2/y1), s1(^x1x2/y1), s1(x1^x2/y1), s0(x1x2/y2)}.

Когда выходным сигналам автомата Мили удается поставить в соответствие внутренние состояния (по аналогии с автоматами Мура), то описание может быть упрощено. Фактически в этом случае идет речь о смешанной модели автоматов Мили-Мура. Для приведенного выше примера состоянию s1 можно поставить в соответствие сигнал y1, а состоянию s0 - y2. В этом случае функция F (3) будет следующей:

(4) F:
s0 = {s1(^x1^x2/-), s1(^x1x2/-), s1(x1^x2/-), s0(x1x2/-)},
s1 = {s1(^x1^x2/-), s1(^x1x2/-), s1(x1^x2/-), s0(x1x2/-)}.

В приведенных примерах F является вполне определенным структурным автоматом. Начиная с формулы (4), можно уже выделить упомянутые выше автоматы S и ^S. Обозначив их соответственно F и ^F, получим отображения следующего вида:

(5) F:
s0 = {s1(^x1^x2/-), s1(^x1x2/-), s1(x1^x2/-)},
s1 = {s0(x1x2/-)},

(6) ^F:
s0 = {s0(x1x2/-)},
s1 = {s1(^x1^x2/-), s1(^x1x2/-), s1(x1^x2/-)}.

Далее отображение M любого структурного автомата будем ассоциировать с формой ДНФ СКА, т.е. с отображением S (см. (1)). Это справедливо в силу того, что подобно нормальным формам булевых функций по S можно восстановить отображение ^S и соответственно исходный вполне определенный автомат M.

6. Минимизация ДКФ СКА

Дизъюнктивные канонические формы структурных конечных автоматов (ДКФ СКА) после выполнения операций минимизации переходят в эквивалентные, но более компактные, формы ДНФ СКА.

Минимизации подлежат ДКФ булевых функций, построенных из конституент единиц переходов, имеющих общие входное и выходное состояния и помеченных одинаковыми наборами выходных сигналов или символом прочерк.

Для рассматриваемого примера необходимо выполнить минимизацию булевой функции:

(7) y = ^x1^x2 ⋁ ^x1x2 ⋁ x1^x2 = ^x1 ⋁ ^x2.

В результате отображения F и ^F, представленные формулами (5) и (6), в минимизированной нормальной форме примут вид:

(8) F:

s0 = {s1(^x1/-), s1(^x2/-)},
s1 = {s0(x1x2/-)}.

(9) ^F:

s0 = {s0(x1x2/-)},
s1 = {s1(^x1/-), s1(^x2/-)}.

Замечание 1. По ДНФ СКА, выполняя действия обратные рассмотренной процедуре минимизации автомата, находится каноническое представление автомата - ДКФ СКА. Далее легко найти дополнение до вполне определенного автомата и, следовательно, восстановить исходное представление любого вполне определенного автомата.

Таким образом, любой вполне определенный дизъюнктивный структурный автомат можно представить дизъюнктивной нормальной формой. Так, для автомата в форме ДКФ СКА, представленного формулой (2), это будет ДНФ СКА, представленный формулой (8).

Простое сравнение формул (2) и (8) демонстрирует компактность дизъюнктивной формы описания автоматов в сравнении с обычной формой. Форма ДНФ СКА дает более ясное и четкое представление о функционировании автомата. Так, например, по виду (8) можно сразу отделить входные ситуации, влияющие на поведение автомата, от ситуаций, не изменяющих его состояние.

7. Сети дизъюнктивных автоматов

Синхронные сети (схемы) из структурных автоматов (ССА), описанные в [6], представляют достаточно простую и адекватную реальным физическим процессам формальную модель. При этом в рамках определения ССА в качестве модели компонентного автомата сети могут быть выбраны ДНФ СКА.

Замечание 2. Если говорить об алгоритмической модели параллельных процессов, то ССА является лишь частью более общей модели процессов – параллельной алгоритмической машины Мили. Последняя, напомним, кроме автоматного управления (в форме ДНФ СКА или ССА на базе ДНФ СКА), включает дополнительно модели памяти и операторов.

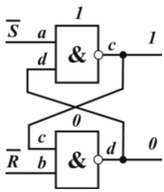

Рис. 1. Схема RS-триггера на элементах И-НЕ

Учитывая, что ДНФ СКА, представленный формулой (7), представляет собой модель элемента И-НЕ, рассмотрим модель асинхронного RS-триггера (рис.1) в форме ССА, состоящей из двух подобных автоматов. Ее описание в аналитической форме ДНФ СКА будет выглядеть следующим образом:

(10) S:

$$s0 = \{s1(\wedge a/-), s1(\wedge d/-)\},$$
$$s1 = \{s0(ad/-)\}.$$

R:

$$w0 = \{w1(\wedge c/-), w1(\wedge b/-)\},$$
$$w1 = \{w0(cb/-)\}.$$

Здесь логическим переменным a, b, c, d (см. также рис.1) соответствуют входные сигналы, принадлежащие соответствующим компонентным автоматам сети. Можно также говорить, что в этом случае они являются локальными переменными соответствующих структурных автоматов.

Введем глобальные переменные x1 и x2, соответствующие установочным входам RS-триггера, и свяжем внутренние состояния автоматных моделей элементов И-НЕ с состоянием его выходов. В результате модель RS-триггера (10) можно представить в следующем эквивалентном виде:

(11) S:

$$s1 = \{s0(x1w1/-)\},$$
$$s0 = \{s1(\wedge x1/-), s1(w0/-)\}.$$

R:

$$w1 = \{w0(s1x2/-)\},$$
$$w0 = \{w1(s0/-), w1(\wedge x2/-)\}.$$

Аналитическая форма описания удобна для различных математических преобразований, но более наглядна форма автоматных графов. На рис.2 показана эквивалентная аналитической форме графическая модель RS-триггера, дополненная с целью представления связей и взаимодействия компонентных автоматов сети элементами, заимствованными у сетей Петри.

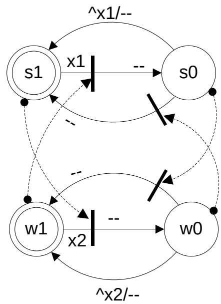

Рис. 2. Граф сетевой модели RS-триггера на элементах И-НЕ

На графе сетевой автоматной модели двойными кружками выделены начальные состояния компонентных автоматов сети. Штрихпунктирные дуги представляют *синхронизирующие связи*, представленные на уровне отдельного структурного автомате его входными сигналами.

Синхронизирующие связи направлены от одних компонентных автоматов к другим автоматам сети. При этом внутренние состояния первых являются источниками входных сигналов для вторых. «Синхронизирующий входной сигнал» автомата примет значение true, если текущее состояние <u>другого</u> автомата примет соответствующее значение, и false в противном случае, т.е. при нахождении <u>другого</u> автомата в любом другом состоянии.

8. Операция композиции

Полное представление об алгоритме функционирования модели параллельного типа дает эквивалентный сети однокомпонентный автомат. В основе его нахождения лежит алгебраическая операция *композиции* автоматов. Ее определение для абстрактных автоматов дано в [9]. Возьмем данное определение за основу и введем аналогичную операцию для структурных автоматов в форме ДНФ СКА.

Для автоматов в форме ДНФ СКА операция *композиции*, обозначаемая далее как $\circ$, определяется следующим образом. Пусть M – бесконечное множество структурных автоматов и A и $B \in M$ – произвольные непустые автоматы в форме ДНФ СКА. Обозначим их через $A = (X, V, L, v_1 \in V, F(x \in X, w \in W'/l \in L))$ и $B = (Y, W, T, w_1 \in W, P(y \in Y, q \in Q'/t \in T))$, где $X, L\, и\, Y, T$ – соответственно входные и выходные структурные алфавиты, $Q\, и\, W$ – алфавиты состояний, $Q^{'} \subseteq Q, W' \subseteq W$– подмножества состояний, используемые для синхронизации автоматов, являющиеся в этом случае входными сигналами автоматов, а $F\, и\, P$ – отображения соответственно Q и W в себя. ДНФ СКА C$= (Z, Q, M, q_1 \in Q, K(z \in Z/m \in M))$, обозначаемый $\mathrm{C} = A \circ B$, называется композицией автоматов $A\, и\, B$, если

(12) $Z = (X \times W^{'}) \times (Y \times Q^{'}) = X \times Y,$

(13) $Q = V \times W,$

(14) $M = L \times T,$

(15) $Kq = F_w v \times P_v w,$

где $q = (v, w), z = (x, y), m = (l, t)$, а $F_w v$ – отображение автомата состояний v автомата A по конъюнкциям входных логических переменных из элементов множества X и подмножества состояний автомата B $W' \subseteq W$, используемых для синхронизации работы автоматов, $P_v w$ – отображение автомата состояний w автомата B по конъюнкциям входных логических переменных из элементов множества Y и подмножества состояний автомата A $V' \subseteq V$, используемых для синхронизации работы автоматов. Начальным состоянием автомата C будет состояние $q_1 = (v_1, w_1)$.

Поскольку $A\, и\, B$ вполне определенные автоматы, то и автомат C будет вполне определенным автоматом.

С учетом особенностей структурных автоматов, специфики использования в качестве синхронизирующих информационных связей информации о текущих внутренних состояниях автоматов, структурная схема автомата С, эквивалентного совместной работе автоматов А и В, показана на рис.3.

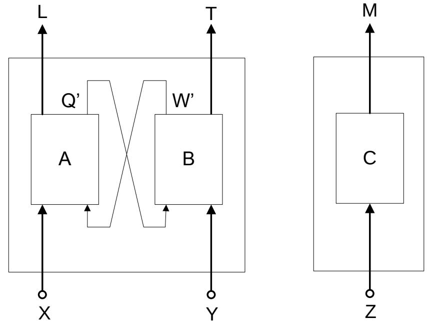

Рис. 3. Представление композиции автоматов

Исходя из определения ДНФ СКА и выражения (15), вычисление результирующего отображения K можно упростить, исключив операцию умножения дополнений автоматов. Таким образом, вычисление отображения R можно представить объединением произведения отображений в виде следующей формулы:

(16) $$Kq = F_w v \times P_v w = (F \times P) \cup ({}^\wedge F \times P) \cup (F \times {}^\wedge P)$$

С точки зрения функционирования компонентных автоматов элементы формулы (16) отражают одновременную работу компонентных автоматов А и В или каждого из них по отдельности, когда другой находится в состоянии «спячки». При этом активными будут автоматы, представленные отображениями в форме ДНФ СКА, т.е. соответственно отображениями F или P.

Использование внутренних состояний для синхронизации автоматов позволяет упростить процедуру вычисления композиции. В этом случае из вычислений исключаются элементы с несовпадающими именами входных логических переменных, представленных именами внутренних состояний, в правой части, с именами внутренних состояний компонентных автоматов в левой части формулы (16).

Покажем применение операции композиции на примере вычисления результирующего автомата для модели RS-триггера (11). С учетом дополнений (9) для каждого из компонентных автоматов R и S вычисление их композиции, т.е. отображения представленного формулой (16), будет выглядеть следующим образом.

(17) $RS = R \circ S = (R \times S) \cup (R \times {}^\wedge S) \cup ({}^\wedge R \times S)$

Дополнения отображений R и S в форме ДНФ СКА до вполне определенных автоматов будут следующими (см. также рис.2):

(18) ^R:

w0 = {w0(s1x2/-)},
w1 = {w1(s0/-), w1(^x2/-)}.

(19) ^S:

s0 = {s0(x1w1/-)},
s1 = {s1(w0/-), s1(^x2/-)}.

Рассмотрим вычисление множества переходов для компонентных состояний {w0s0, w1s0, w0s1, w1s1} подавтомата R× S (см. также (11)):

(20) $R \times S$:

$w0s0 = \{w1s1({}^\wedge x1), w1s1(-), w1s1({}^\wedge x1{}^\wedge x2), w1s1({}^\wedge x2)\} = \{w1s1(-)\}$.

Вид формулы (20) следует из того, что переход $w1s1(-)$ является безусловным и фактически покрывает любые другие переходы.

(21) $w1s0 = \{w0s1({}^\wedge x1s1x2), w0s1(w0s1x2)\} = \{-\}$.

В формуле (21) переходы, входные конъюнкции которых содержат состояние s1, исключены из рассмотрения, т.к. текущее компонентное состояние содержит состояние s0. Поскольку других переходов для компонентного состояния $w1s0$ у подавтомата $R \times S$ нет, то и получаем в результате пустое множество переходов.

Далее по аналогии получим множества переходов для остальных компонентных состояний:

(22) $w0s1 = \{w1s0(x1w1s0), w1s0(x1w1{}^\wedge x2)\} = \{-\}$.

(23) $w1s1 = \{w0s0(x1w1s1x2)\} = \{w0s0(x1x2)\}$.

Для формулы (22) переходы определены аналогично вычислению переходов для формулы (21), где: справа – s0, слева – s1; справа – w1, слева – w0. Для формулы (23) входная дизъюнкция вида x1w1s1x2 упрощена до x1x2, т.к. элементы состояний текущего компонентного состояния совпадают с именами логических переменных внутренних состояний, включенных в условие перехода.

В итоге подавтомат $R \times S$, определяющий одновременную работу автоматов композиции, содержит фактически всего два перехода:

$R \times S$:

$$w0s0 = \{w1s1(-)\},$$
$$w1s1 = \{w0s0(x1x2)\}.$$

По аналогии для подавтомата $\hat{}R \times S$ переходы для его компонентных состояний w0s0, w0s1, w1s0 и w1s1 будут следующими:

$w0s0 = \{-\}$, т.к. ^R содержит единственный переход при условии s1, а рассматривается s0;

$w0s1 = \{-\}$, т.к. S сдержит единственный переход для w1, а рассматриваемое состояние w0;

$w1s0 =$
$\{s1w1(\hat{}x1), -, -, w1s1(\hat{}x1\hat{}x2), -, w1s0(x1\hat{}x2)\} =$
$\{s1w1(\hat{}x1)\}$,

т.к. переход $s1w1(\hat{}x1)$ поглощает $w1s1(\hat{}x1\hat{}x2)$, а переход $w1s0(x1\hat{}x2)$ не изменяет текущее состояние и не содержит выходного сигнала;

$w1s1 = \{-, w1s0(\hat{}x2)\} = \{w1s0(\hat{}x2)\}$.

Таким образом:

$\hat{}R \times S$:

$$w1s0 = \{w1s1(\hat{}x1)\},$$
$$w1s1 = \{w1s0(x1\hat{}x2)\}.$$

Проведя аналогичные действия, получим:

$R \times \hat{}S$:

$$w0s1 = \{w1s1(\hat{}x2)\},$$
$$w1s1 = \{w0s1(\hat{}x1x2)\},$$

Объединив в соответствии с (16) вычисленные подавтоматы, получим эквивалентный сетевой модели однокомпонентный автомат:

RS:

$w0s0 = \{s1w1(x1x2), w0s1(\hat{}x1x2), w1s0(x1\hat{}x2)\}$,

$w0s1 = \{s1w1(\hat{}x2)\}$,

$w1s0 = \{s1w1(\hat{}x1)\}$,

$w1s1 = \{w0s0(x1x2), w1s0(x1\hat{}x2), w0s1(\hat{}x1x2)\}$.

Граф композиции автоматов R и S, т.е. эквивалентного автомата для сетевой модели RS-триггера, показан на рис.4.

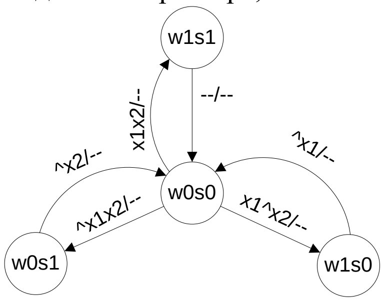

Рис. 4. Граф результирующего автомата модели RS-триггера

На графе результирующего автомата особенности функционирования RS-триггера становятся очевидными. Например, ясно, что переход из одного устойчивого состояние триггера в другое **обязательно** происходит через так называемое *запрещенное* состояние. Вероятен и совсем парадоксальный для триггера случай - режим *генерации*. В реальной жизни он невозможен лишь по причине того, что реальные элементы триггера, в отличие от своих моделей, не могут быть абсолютно одинаковыми, а отличаются друг от друга, как минимум, величиной задержек.

9. Автоматная модель пункта обслуживания

В противоположность утверждению, высказанному в отношении автоматов в [10], сравним возможности описания параллельных процессов сетевой автоматной моделью с моделью сетей Петри.

Сделаем это на примере часто рассматриваемой в сетях Петри модели системы, ассоциируемой с пунктом обслуживания, которым может быть достаточно широкий класс реальных и формальных объектов[5]. Например, это может быть продавец, ожидающий покупателя[10].

Сеть Петри, представляющая модель пункта обслуживания, показана на рис. 5. На ней фишка в позиции p1 означает появление запроса на обслуживание, а появление фишки в позиции p4 – успешное его выполнение.

Последовательность смены маркировок сети Петри, отражающая ее работу, показана на рис. 6.

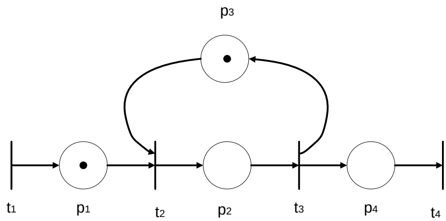

Рис.5. Сеть Петри для пункта обслуживания

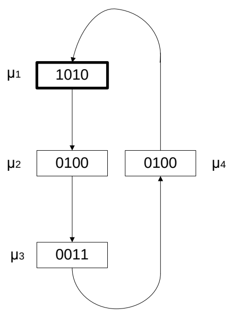

Рис.6. Последовательность маркировок сети Петри

Создадим сеть из автоматов в форме ДНФ СКА, представляющую параллельную автоматную модель аналогичного пункта обслуживания. Она может выглядеть так, как показано на рис.7, где автомат S реализует обслуживание, а автомат W –

синхронизирует процесс обслуживания с появлением запроса на него.

Для автомата S истинное значение входного сигнала x1 соответствует появлению запроса, а выдача выходного сигнала y1 – реализации процедуры обслуживания. При этом переход из состояния ожидания обслуживания s1 в состояние реализации обслуживания s2 возможен только, когда есть запрос на обслуживание (x1 – true), а автомат W находится в состоянии g, разрешая обслуживание запроса.

Переход автомата W из состояния g в r запрещает обслуживание поступившего запроса до момента завершения обслуживания текущего запроса, т.е. до перехода автомата S в состояние s2.

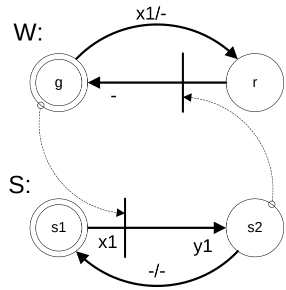

Рис.7. Автоматная модель пункта обслуживания

Определим поведение модели автоматного пункта обслуживания по эквивалентному сети автомату, используя для его нахождения операцию композиции автоматов.

В аналитической форме модель автоматной сети пункта обслуживания, соответствующая графу на рис.7, будет следующей:

W:	^W:
g = {r(x1/-)},	g = {-},
r = {g(s2/-)}.	r = {r(^s2/-)}.
S:	^S:
s1 = {s2(gx1/-)},	s1 = {s1(^x1/-), s1(^g/-)},
s2 = {s1(-/-)}.	s2 = {-}.

К = W∘S = W×S∪^W×S∪W×^S, где

W×S:

gs1 = {rs2(gx1/-)} = {rs2(x1/-)},

gs2 = {rs1(x1/-)},

rs1 = {-},

rs2 = {gs1(-/-)}.

^W×S:

gs1 = {-},

gs2 = {-},

rs1 = {-},

rs2 = {-}.

W×^S:

gs1 = {-},

gs2 = {-},

rs1 = {-},

rs2 = {-}.

После объединения подавтоматов получим результирующий автомат К:

К = W∘S:

gs1 = {rs2(x1/-)},

gs2 = {rs1(x1/-), gs1(^x1/-)},

rs1 = {-},

rs2 = {gs1(^x1/-)}.

Граф автомата К показан на рис.8.

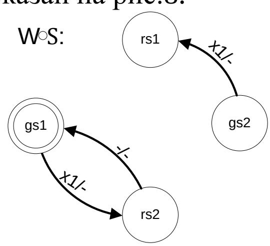

Рис.8. Эквивалентный автомат для сети автоматов рис.7

После исключения состояний gs2 и rs1, которые недоступны из начального состояния автоматной сети gs1, и переименования состояний результирующий автомат сети сводится к автомату на рис. 9. Где состояния green и red представляют «зеленое» и «красное» компонентные состояния сети, которые разрешают и, соответственно, запрещают доступ к пункту обслуживания.

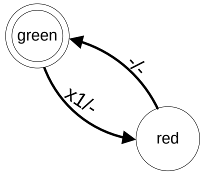

Рис.9. Автомат эквивалентный работе автоматной сети рис.8

Рассмотрим ситуацию, когда модель пункта обслуживания представлена «ошибочной» автоматной сетью, показанной на рис. 10.

Найдем эквивалентный автомат, используя в этот раз «ручной» алгоритм нахождения эквивалентного автомата по графу сети. Для чего: 1) зафиксируем все множество компонентных состояний системы и 2) для каждого такого состояния на основе визуального анализа соответствующих состояний компонентных автоматов сети найдем переходы в другие состояния.

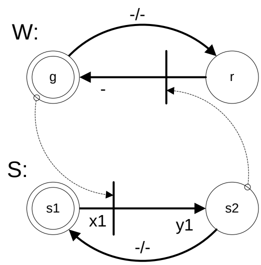

Рис.10. Еще одна автоматная модель пункта обслуживания

Например, для компонентного состояния gs1 (состояний g и s1 соответственно автоматов W и S) будет два перехода - в состояния rs2 и rs1 в зависимости от значения x1. Это следует из того, что автомат S перейдет из состояния s1 в s2 или останется в нем, а автомат W в любой ситуации перейдет из состояния g в состояние r.

Проанализировав остальные парные состояния сети, получим граф автомата, показанный на рис. 11.

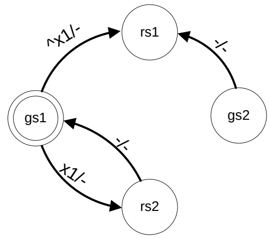

Рис.11. Автомат, эквивалентный сети на рис.10

Из него следует, что сеть на рис. 11 в определенной ситуации может перейти в тупиковое состояние rs1, из которого уже нет переходов в другие состояния.

Таким образом, модель пункта обслуживания, предложенная на рис. 10, фактически неработоспособна. Кроме, правда, ситуации, когда запросы на обслуживание идут непрерывным потоком с минимальными паузами между ними. Причем следующий запрос на обслуживание по отношению к текущему запросу должен поступить до перехода сети из состояния rs2 в gs1 (см. рис.10).

На примере автоматной модели пункта обслуживания (точнее, двух его моделей – правильной и с ошибкой) были показаны возможности анализа систем на базе операции композиции автоматов и обоснована ее необходимость для строго доказательства свойств автоматной сети. В результате то, что не столь очевидно при сравнении сетевых моделей, приобретает явные черты при сравнении эквивалентных им однокомпонентных моделей.

10. Заключение

Теория автоматов в классическом варианте ориентирована на прямое проектирование систем. В этом случае строится однокомпонентная модель абстрактного автомата, а затем рассматривается ее реализация в форме схемы, представленной структурным автоматом. Это не может не вызвать проблемы в приложении к сложным многокомпонентным системам. Хотя бы просто потому, что создание однокомпонентной модели для изначально многокомпонентной системы, определяемой огромным числом ее состояний, представляет, порой, неразрешимую задачу.

Но, если не ставится та же задача строго доказательства алгоритмических свойств сложной системы (см. результирующие автоматы RS-триггера и пункта обслуживания), то, как правило, в однокомпонентной модели нет необходимости. Формулировки структурной модели автомата в «абстрактной форме», использование автоматных схем для описания изначально многокомпонентных систем позволяет распространить теорию структурных автоматов и на макроуровень проектирования многокомпонентных систем.

В работе рассмотрены общие проблемы теории автоматов, которые сдерживают применение автоматных моделей применительно к сложным многокомпонентным системам параллельного типа. Предложенная модель отдельной компоненты и системы в целом, изменения в законе функционирования автоматов, более четкое определение автоматов, как модели управления, одни из путей, расширяющих область применения теории автоматов на область больших систем.

Дано определение модели конечного структурного автомата в форме ДНФ СКА. Для большинства реальных моделей она дает более наглядное, ясное и компактное описание, т.к. переходы, помеченные элементарными конъюнкциями ранга меньшего, чем m – числа входов структурного автомата, покрывают соответствующее множество переходов.

Введенная модель структурного автомата предложена в качестве основы для создания простой, наглядной и логично построенной [алгоритмической] параллельной модели, теория которой, кроме уже существующей теории автоматов, может быть дополнена алгеброй структурных автоматов.

В работе приведено описание «автоматной машины Мили», снимающей все ограничения на вычислительные возможности автоматов. В качестве конечно-автоматной модели управления абстрактной алгоритмической машины автоматы смотрятся более чем убедительно. И то, как они при этом решают проблемы описания, синтеза и анализа параллельных систем, позавидуют любые известные алгоритмические [параллельные] модели.

Показано, что исключение понятия «мгновенных» автоматов, позволяет упростить модель и решить проблему циклических цепей для сложных систем. Сетевая автоматная модель стала более адекватна структуре и функционированию реальным объектам. А смоделировать любое относительное быстродействие, как схем, так и отдельных элементов, легко, если рассматривать их функционирование в разных дискретных временах, задавая сколь угодно малое реальное время дискретному такту схеме в целом или ее отдельному элементу.

На конкретных примерах в работе дано сравнение предложенной параллельной модели, представленной сетью из структурных автоматов, с другими параллельными моделями (на базе сетей Петри). Показано применение введенной операции композиции для анализа параллельных систем на примерах создания и анализа моделей RS-триггера и модели пункта обслуживания.

Направлением дальнейших работ по сетям на базе автоматов в форме ДНФ СКА может стать определение операции *декомпозиции*. Решение данной задачи для произвольных автоматов фактически решает проблему автоматического распараллеливания программ в общем случае, поскольку с точки зрения теории автоматов ее решение может быть сведено к проблеме синтеза цифровых схем. Канонический метод структурного синтеза автоматов – фактически одно из уже существующих на эту тему решений.

Литература

1. ГЛУШКОВ В.М. *Введение в кибернетику*. Изд-во АН Украинской ССР. К.: 1964. - 324с.
2. ГЛУШКОВ В.М. *Синтез цифровых автоматов.* М.: Физматгиз, 1962.
3. ГЛУШКОВ В.М., ЦЕЙТЛИН Г.Е., ЮЩЕНКО Е.Л. *Алгебра. Языки. Программирование. (изд. 2-е, перераб.).* – К., 1978. – 320 с.
4. КУДРЯВЦЕВ В.Б., АЛЕШИН С.В., ПЛДКОЛЗИН А.С. *Введение в теорию автоматов* – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 320 с.

5. КУЗНЕЦОВ О.П. *Лекция 6. Алгоритмические возможности конечных автоматов. Сети Петри.* URL: http://www.intuit.ru/studies/courses/555/411/lecture/9429 (дата обращения: 03.03.16).
6. КУЗНЕЦОВ О.П., АНДЕЛЬСОН-ВЕЛЬСКИЙ Г.М. *Дискретная математика для инженера.* – 2-е изд., перераб. и доп. – М.: Энергоатом издат, 1988. – 480 с.
7. 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.
8. *Математическая энциклопедия.* – М.: Сов. энциклопедия, Т.1. Статья: «Автомат конечный», (с.52 – 67).
9. МЕЛИХОВ А.Н. *Ориентированные графы и конечные автоматы.* – М.: Наука, 1971. – 416 с.
10. ПИТЕРСОН Дж. *Теория сетей Петри и моделирование систем:* Пер с англ. – М.: Мир, 1984. – 264с.

ARTICLE TITLE

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

Abstract: The abstract model of calculations in the automatic form is considered and the disjunctive form of representation of structural automatic machines is formulated. The network model and algebraic operation of a composition on the basis of structural automatic machines is offered. The automatic model of management of parallel calculations is described. Comparison with networks Petri is resulted.

Keywords: the theory of finite-state machines, a disjunctive normal form of structural finite-state machines, operation of a composition of finite-state machines, parallel model of calculations.

Рубрика Сборника (*окончательно выбирается редактором*)

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

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