

РАДИОФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

Н.В. Евтушенко, М.В. Рекун,
С.В. Тихомирова

**НЕДЕТЕРМИНИРОВАННЫЕ
АВТОМАТЫ: АНАЛИЗ И СИНТЕЗ**

Часть 2
**РЕШЕНИЕ АВТОМАТНЫХ
УРАВНЕНИЙ**



ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Н.В. Евтушенко, М.В. Рекун, С.В. Тихомирова

**НЕДЕТЕРМИНИРОВАННЫЕ АВТОМАТЫ:
АНАЛИЗ И СИНТЕЗ**

ЧАСТЬ 2. РЕШЕНИЕ АВТОМАТНЫХ УРАВНЕНИЙ

Томск 2009

УДК 519.713

Н.В. Евтушенко, М.В. Рекун, С.В. Тихомирова. Недетерминированные автоматы: анализ и синтез. Ч. 2. Решение автоматных уравнений: Учебное пособие. – Томск: Томский государственный университет, 2009. – 111 с. – 100 экз.

Настоящее пособие является второй частью монографии и посвящено решению уравнений и неравенств в алгебре конечных автоматов. Авторы обсуждают различные области применения автоматных уравнений, методы и алгоритмы решения таких уравнений. Пособие предназначено для студентов кафедры информационных технологий и исследования дискретных структур (ИТИДиС) РФФ ТГУ, изучающие курсы «Теория автоматов» и «Техническая диагностика», а также для всех интересующихся данной темой.

Работа выполнена при поддержке Грантов РФФИ, Минобразования РФ, НАТО.

Рецензент – д.т.н. А.Ю. Матросова.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1. ЗАДАЧИ, В КОТОРЫХ ИСПОЛЬЗУЕТСЯ РЕШЕНИЕ АВТОМАТНЫХ УРАВНЕНИЙ	7
1.1. КОНЕЧНЫЕ АВТОМАТЫ	7
1.2. КОНЕЧНО АВТОМАТНЫЕ ОТОБРАЖЕНИЯ, ЯЗЫКИ И ОТНОШЕНИЯ МЕЖДУ АВТОМАТАМИ.....	12
1.3. ПРЕДСТАВЛЕНИЕ АВТОМАТОВ ПОЛУАВТОМАТАМИ	17
1.4. КОМПОЗИЦИЯ АВТОМАТОВ.....	19
1.4.1. Синхронная композиция конечных автоматов	20
1.4.2. Параллельная композиция конечных автоматов	24
1.5. АВТОМАТНЫЕ НЕРАВЕНСТВА И УРАВНЕНИЯ.....	29
1.6. РЕШЕНИЕ АВТОМАТНЫХ УРАВНЕНИЙ В РАЗЛИЧНЫХ ПРИЛОЖЕНИЯХ	31
1.6.1. Оптимизация цифровых схем	32
1.6.2. Логический синтез	34
1.6.3. Тестирование и диагностика дискретных систем.....	35
1.7. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 1	37
2. РЕШЕНИЕ СИНХРОННОГО АВТОМАТНОГО УРАВНЕНИЯ	39
2.1. РЕШЕНИЕ АВТОМАТНОГО УРАВНЕНИЯ ДЛЯ ПОСЛЕДОВАТЕЛЬНОЙ КОМПОЗИЦИИ	39
2.1.1. Решение автоматного уравнения для хвостовой компоненты последовательной композиции	39
2.1.2. Решение автоматного уравнения для головной компоненты последовательной композиции	42
2.2. РЕШЕНИЕ АВТОМАТНОГО УРАВНЕНИЯ ДЛЯ КОМПОЗИЦИИ С ОБРАТНЫМИ СВЯЗЯМИ	46
2.2.1. Языковой подход к решению автоматного уравнения	46
2.2.2. Алгоритм построения наибольшего решения в наибольшем алфавите	51
2.2.3. Решение автоматного уравнения на основе безразличных последовательностей	54
2.3. ЧАСТНЫЕ РЕШЕНИЯ СИНХРОННОГО АВТОМАТНОГО УРАВНЕНИЯ.....	58

2.3.1. Полностью определенные решения синхронного автоматного уравнения	59
2.3.2. Муровские решения синхронного автоматного уравнения.....	61
2.3.3. Другие частные решения синхронного автоматного уравнения	63
2.4. РАЗРЕШИМОСТЬ СИНХРОННОГО АВТОМАТНОГО УРАВНЕНИЯ ОТНОСИТЕЛЬНО РАЗЛИЧНЫХ ТОПОЛОГИЙ	65
2.5. СИСТЕМЫ СИНХРОННЫХ АВТОМАТНЫХ УРАВНЕНИЙ.....	67
2.5.1. Системы синхронных автоматных уравнений при локальной оптимизации автоматных сетей.....	68
2.5.2. Решение системы синхронных автоматных уравнений ..	70
2.5.3. Частные решения системы автоматных уравнений.....	72
2.6. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 2	73
3. РЕШЕНИЕ ПАРАЛЛЕЛЬНОГО АВТОМАТНОГО УРАВНЕНИЯ.....	75
3.1. РЕШЕНИЕ ПАРАЛЛЕЛЬНОГО АВТОМАТНОГО УРАВНЕНИЯ	75
3.2. L-ОГРАНИЧЕННЫЕ РЕШЕНИЯ.....	82
3.3. ПРОГРЕССИВНЫЕ РЕШЕНИЯ ПАРАЛЛЕЛЬНОГО АВТОМАТНОГО УРАВНЕНИЯ	88
3.4. РЕШЕНИЕ СИСТЕМЫ ПАРАЛЛЕЛЬНЫХ АВТОМАТНЫХ УРАВНЕНИЙ.....	90
3.5. КОНТРОЛЬНЫЕ ВОПРОСЫ К ГЛАВЕ 3	93
4. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ АВТОМАТНЫХ УРАВНЕНИЙ В ЛОГИЧЕСКОМ СИНТЕЗЕ	95
4.1 ОПТИМИЗАЦИЯ АВТОМАТНОЙ СЕТИ НА ОСНОВЕ РЕШЕНИЯ УРАВНЕНИЙ	95
4.2. ОПТИМИЗАЦИЯ КОМБИНАЦИОННОЙ СХЕМЫ НА ОСНОВЕ РЕШЕНИЯ УРАВНЕНИЙ.....	97
4.3. ПРИМЕР ПРОТОКОЛЬНОГО КОНВЕРТОРА НА ОСНОВЕ РЕШЕНИЯ УРАВНЕНИЙ.....	104
ЛИТЕРАТУРА	109

ВВЕДЕНИЕ

Вторая часть настоящего пособия посвящена решению уравнений в алгебре конечных автоматов. Сложные управляющие системы, в частности цифровые схемы, обычно являются многоэлементными, т.е. могут быть представлены в виде сети взаимодействующих компонент. Хорошо известно, что в качестве одной и той же компоненты можно использовать различные логические подсистемы, и при этом внешнее поведение системы не изменится. Естественным образом возникает вопрос, каким образом допустимо изменять поведение компоненты в зависимости от остальных (фиксированных) компонент и известной спецификации всей системы. Вопрос возникает в ряде приложений, таких как оптимизация цифровых схем, логический синтез, криптография (автоматные криптосистемы), логические игры (формирование выигрышной или беспроигрышной стратегии), синтез стабилизаторов для асинхронных систем, и др. Для формального решения задачи необходимо ответить на следующие вопросы.

1. Какая формальная модель используется для описания поведения компонент и системы в целом?
2. С какой точностью синтезированная система должна соответствовать спецификации?
3. Каковы правила совместного функционирования элементов системы (синхронные, асинхронные и т.п.)?

Если по всем этим вопросам решение принято, то проблема сводится к решению уравнения $A @ X \sim S$, где X – интересующая нас компонента, контекст A описывает поведение известной части системы, S – спецификация системы, $@$ – операция композиции элементов системы, и \sim – отношение конформности, в котором должны находиться синтезируемая система и ее спецификация. Для различных приложений эти параметры определяются по-разному.

В данном пособии мы полагаем, что поведение всех систем описано конечными автоматами, и мы остановимся на решении автоматных уравнений относительно двух операций композиции, синхронной и параллельной. При синхронном функционировании в каждый такт активными являются обе компоненты; композиция производит внешний выходной сигнал и согласованную пару внутренних выходных символов. Такая композиция, например, используется при описании поведения многоэлементных цифровых схем. В случае параллельной композиции в каждый такт активной является только одна компонента,

и внешний сигнал появляется на выходе композиции только после окончания внутреннего диалога между компонентами. Такая композиция используется, в частности, в телекоммуникационных системах. Мы также рассматриваем два отношения конформности: отношение редукции и отношение эквивалентности. Автомат B является редукцией автомата A , если поведение автомата B содержится в поведении автомата A . Автоматы A и B эквивалентны, если эти автоматы имеют одинаковое поведение. Отношение редукции является отношением порядка на языках конечных автоматов; поэтому можно говорить об автоматных синхронных и параллельных неравенствах и уравнениях.

Чтобы сделать эту часть пособия максимально независимой, в первой главе мы напоминаем читателю основные определения из теории автоматов, в том числе определения и обозначения для недетерминированных автоматов, для синхронной и параллельной композиции автоматов, а также вводим понятия синхронного и параллельного автоматных уравнений и неравенств. Во второй части первой главы мы обсуждаем приложения, в которых используются автоматные уравнения.

Вторая глава посвящена решению синхронных автоматных неравенств и уравнений. В частности, мы показываем, что разрешимое синхронное автоматное неравенство всегда имеет наибольшее (общее) решение, которое содержит в себе множество всех решений неравенства. В третьей главе рассматривается решение параллельных автоматных неравенств и уравнений. Разрешимое параллельное автоматное неравенство также всегда имеет наибольшее (общее) решение. Мы вводим и исследуем некоторые частные решения автоматных уравнений, которые являются интересными с теоретической/практической точки зрения.

В четвертой главе приводятся примеры использования автоматных уравнений. В частности, на основе приведенных в пособии методов и алгоритмов было создано программное обеспечение для оптимизации цифровых схем. Проведенные нами компьютерные эксперименты на контрольных примерах из сети Интернет показывают, что не только игрушечные примеры из четвертой главы, но и реальные цифровые схемы могут быть оптимизированы на основе решения соответствующих автоматных уравнений.

1. ЗАДАЧИ, В КОТОРЫХ ИСПОЛЬЗУЕТСЯ РЕШЕНИЕ АВТОМАТНЫХ УРАВНЕНИЙ

В этой главе мы рассказываем о различных приложениях, в которых может быть использовано и используется решение автоматных уравнений. Мы предполагаем, что читатель знаком с основными понятиями теории автоматов. Тем не менее, для того чтобы сделать эту часть пособия максимально независимой от других частей, в первом разделе мы напоминаем определение автомата и определение отношений эквивалентности и редукции между автоматами. Конечный автомат в этой части пособия используется только для представления специального класса словарных отображений (последовательностных функций), сопоставляющих каждому слову (последовательности) в одном (входном) алфавите одно или несколько слов той же длины в другом (выходном) алфавите. Кроме того, так же как и в первой части пособия, в этой части мы рассматриваем только инициальные автоматы, то есть автоматы, описывающие поведение дискретных систем, обладающих сигналом сброса.

1.1. Конечные автоматы

Конечным автоматом или просто автоматом называется пятерка $A = (A, I, O, T_A, a_0)$, где A – конечное множество состояний с выделенным начальным состоянием a_0 , I – входной алфавит, O – выходной алфавит, и $T_A \subseteq A \times I \times O \times A$ – отношение переходов (рис. 1.1а). Четверка (a, i, o, a') $\in T_A$ описывает переход в автомате A из состояния a в состояние a' под действием входного символа i с выходным символом o . В общем случае, в текущем состоянии для данного входного символа может существовать более одного перехода или не существовать ни одного перехода.

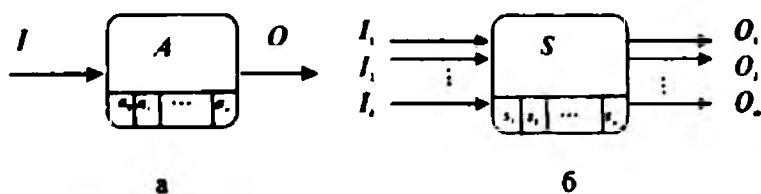


Рис 1.1, а автомат A

б автомат S , входной и выходной алфавиты которого суть декартовы произведения других алфавитов

Иногда входной (выходной) алфавит автомата может быть декартовым произведением нескольких алфавитов; в этом случае говорят о множестве входных (выходных) алфавитов автомата и о входных (выходных) переменных, соответствующих этим алфавитам (рис. 1.16).

Максимальным автоматом (относительно входного алфавита I и выходного алфавита O) с одним состоянием называется автомат $A = (\{a_0\}, I, O, T_A, a_0)$, в котором переходы определены для каждого входного символа со всеми возможными выходными символами, то есть для любых $(i, o) \in I \times O$ ($(a_0, i, o, a_0) \in T_A$). Для краткости мы будем называть такой автомат *максимальным $(I \times O)$ -автоматом*.

Автомат называется *наблюдаемым*, если для любой тройки $(a, i, o) \in A \times I \times O$ существует не более одного состояния $a' \in A$ такого, что $(a, i, o, a') \in T_A$.

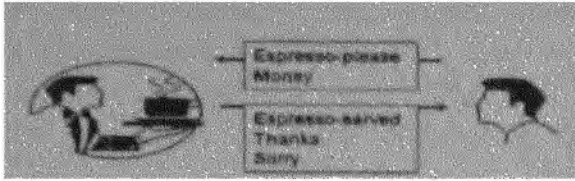
Если для каждой пары $(a, i) \in A \times I$ существует хотя бы один переход $(a, i, o, a') \in T_A$, то автомат называется *полностью определенным*. В противном случае автомат называется *частично определенным* или *частичным*.

Автомат называется *детерминированным*, если для любой пары $(a, i) \in A \times I$ существует не более одной пары $(o, a') \in O \times A$ такой, что $(i, a, a', o) \in T_A$. В противном случае автомат называется *недетерминированным*.

Мы иллюстрируем свойства автоматов на примерах из части I настоящего пособия.

Рассмотрим автомат, описывающий поведение кофейни (рис. 1.2а). Клиент заказывает официанту чашку кофе Ep (*Espresso-please*). В качестве реакции на данное входное воздействие официант приносит посетителю чашку кофе Es (*Espresso-served*). Покупатель платит деньги M (*Money*), официант его благодарит Th (*Thanks*), и кофейня готова обслужить нового посетителя. Если платежеспособность клиента вызывает сомнения у официанта, то официант извиняется S (*Sorry*) и просит сначала оплатить заказ. Автомат, описывающий поведение кофейни (рис. 1.2б), является *недетерминированным*, т.к. возможны различные реакции официанта при заказе кофе (различные реакции на входной символ Ep).¹

¹ Начальное состояние автомата помечается знаком v .



а



б

Рис. 1.2, а кофейня

б автоматное описание кофейни

В качестве следующего примера рассмотрим автомат, продающий билеты на электричку (рис. 1.3а). Билет стоит 4 руб. Пассажир бросает в автомат *монеты* достоинством 1 руб. и 2 руб., т.е. у автомата два входных символа: 1 и 2 (руб.). Когда сумма брошенных монет достигает 4 руб., автомат выдает *билет*. Такое поведение можно описать автоматом на рис. 1.3б. Автомат имеет четыре состояния со следующей семантикой: начальное состояние s_0 – накоплено 0 руб., s_1 – накоплен 1 руб., s_2 – накоплено 2 руб., s_3 – накоплено 3 руб. Выходных символов у автомата два: «пусто» (-), когда автомат не выдает билет, и «билет» (б). Если в начальный момент времени пассажир бросает монету достоинством 2 руб. (входной символ 2), то автомат из начального состояния s_0 переходит в состояние s_2 , имея выходной символ «пусто». Если после этого пассажир бросает монету в 1 руб., то автомат переходит в состояние s_3 , имея выходной символ «пусто». Если пассажир бросает монету достоинством 2 руб. (поступает второй входной символ 2), то автомат выдает билет и возвращается в начальное состояние.² Поскольку поведение автомата однозначно определено в каждом состоянии, то автомат является детерминированным.

Предположим, что автомат по продаже билетов неисправен; например, после выдачи билета «кошелек» автомата не всегда полностью очищается. В этом случае поведение автомата в состояниях s_2 и s_3 перестает быть детерминированным. Если после выдачи билета в «кошелек» автомата останется 1 руб., то автомат из состояний s_2 и s_3 под действием входного символа 2 перейдет в состояние s_1 вместо состояния s_0 . Если после выдачи билета в «кошелек» автомата останется 2 руб., то автомат из состояний s_2 и s_3 под действием входного символа 2 перейдет в состояние s_2 вместо состояния s_0 . Если после выдачи билета «кошелек» автомата полностью очистится, то автомат из состояний s_2 и s_3 под действием входного символа 2 перейдет в ожидаемое состоя-

² Можно предусмотреть возможность выдачи автоматом сдачи

ние s_0 . Такое поведение можно описать недетерминированным автоматом на рис. 1.3в. Автомат на рис. 1.3в является ненаблюдаемым, т.к., например, из состояния s_1 под действием входного символа 2 (руб.) автомат может выдать билет и перейти как в состояние s_0 , так и в состояние s_3 .

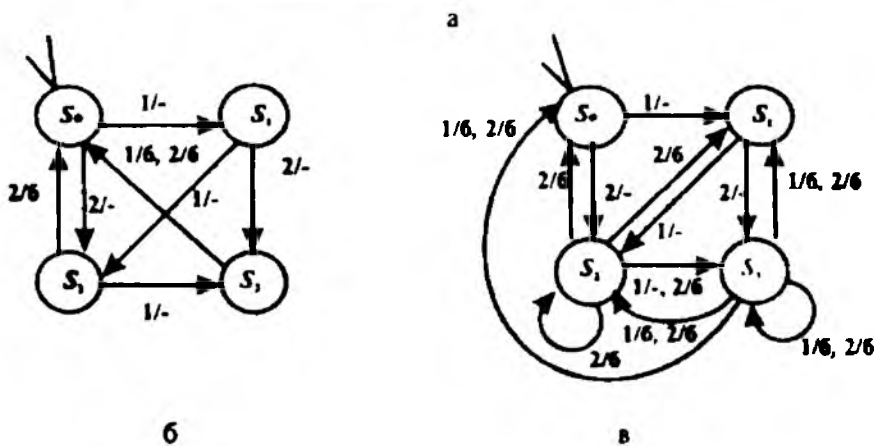
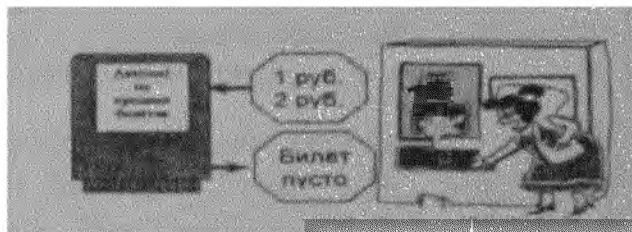
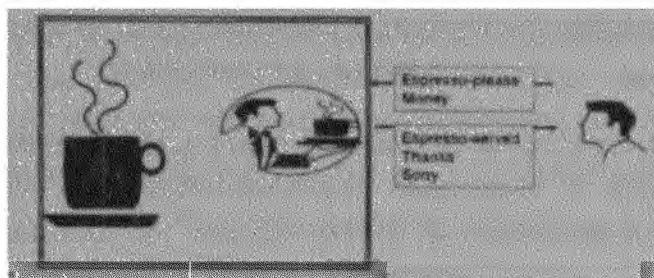


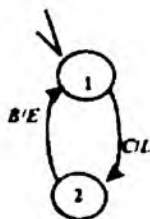
Рис. 1.3. а автомат по продаже билетов на электричку
 б диаграмма переходов автомата по продаже билетов
 в диаграмма переходов неисправного автомата по продаже билетов

Во всех рассмотренных выше примерах автоматы были полностью определенными, т.е. поведение автомата было определено в каждом состоянии для каждого входного символа. Чтобы проиллюстрировать понятие частично определенного (частичного) автомата структурируем далее поведение кофейни. Рассмотрим сеть из двух взаимодействующих автоматов (рис. 1.4а). Первый из них (рис. 1.4б) описывает функционирование автомата по продаже кофе и имеет входной алфавит, состоящий из символов C (*Coin*, жетон), B (*Button*, нажатие кнопки), и выходной алфавит из символов L (*Lamp*, загорается лампочка), E (*Espresso*, выдается кофе). Второй автомат (рис. 1.4в) описывает поведение официанта, который взаимодействует с кофейной машиной посредством сигналов C , B , L , E , и с клиентом посредством сигналов Ep (*Espresso-please*, клиент заказывает чашку кофе), M (*Money*, клиент

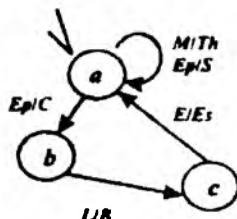
платит деньги), *Es* (*Espresso-served*, официант приносит кофе), *Th* (*Thanks*, официант благодарит) и *S* (*Sorry*, официант извиняется). Клиент, официант и кофейная машина взаимодействуют в режиме диалога. Совместное поведение официанта и кофейной машины соответствует функционированию кофейни и может быть описано автоматом, изображенным на рис. 1.2б.



а



б



в

Рис. 1.4, а кофейня
б кофейная машина
в автомат *Waiter*, описывающий
поведение официанта

Клиент заказывает официанту чашку кофе *Ep* (*Espresso-please*). Если платежеспособность клиента не вызывает сомнений, то в качестве реакции на входное воздействие *Ep* официант опускает жетон *C* (*Coin*) в кофейную машину. Кофейная машина «отвечает» зажиганием лампочки *L* (*Lamp*); в ответ официант нажимает кнопку *B* (*Button*) и получает чашку кофе *E* (*Espresso*), которую подает посетителю *Es* (*Espresso-served*). Если официант не уверен в платежеспособности покупателя, то официант может извиниться *S* (*Sorry*) и попросить клиента сначала оплатить заказ. Таким образом, поведение автомата-официанта (рис. 1.4в) является недетерминированным в состоянии *a*, и, следовательно, недетерминированным является поведение автомата, описывающего поведение кофейни. Клиенту нет смысла платить деньги, когда автомат-официант находится в состояниях *b* и *c*. В этих со-

стояниях официант, «не видит» клиента, т.к. он взаимодействует с кофейной машиной. Таким образом, внешняя среда для официанта является ограниченной, и, как следствие, поведение автомата-официанта не определено, например, в состояниях b и c под действием входного символа M (*Money*), т.е. поведение официанта описывается *частичным недетерминированным* автоматом.

1.2. Конечно автоматные отображения, языки и отношения между автоматами

Отношение переходов автомата A обычным образом распространяется на входные и выходные последовательности. Формально, для входной последовательности $\alpha = i_1 \dots i_k \in I^*$ и выходной последовательности $\beta = o_1 \dots o_k \in O^*$ четверка (a, α, β, a') принадлежит T_A , если и только если существует последовательность состояний $a, a_1, \dots, a_{k-1}, a'$ таких, что $(a, i_1, o_1, a_1) \in T_A, \dots, (a_{k-1}, i_k, o_k, a') \in T_A$. Мы вводим функцию выходов *out* автомата A , которая отображает декартово произведение $A \times I^*$ в множество подмножеств выходных последовательностей. Пусть $\alpha \in I^*$ и $a \in A$. Выходная последовательность β принадлежит $out(a, \alpha)$, если и только если существует состояние a' такое, что четверка (a, α, β, a') принадлежит T_A . В этом случае пара (α, β) называется *входо-выходной* последовательностью автомата в состоянии a . Если $out(a, \alpha) = \emptyset$, то мы говорим, что поведение автомата A в состоянии a не определено на входной последовательности α . По определению, $(a, \varepsilon, \varepsilon, a) \in T_A$, где ε – пустая последовательность, для любого состояния $a \in A$. Поведение полностью определенного автомата в любом состоянии определено на каждой входной последовательности.

Языком автомата A в состоянии a называется множество входо-выходных последовательностей автомата в этом состоянии, т.е. множество последовательностей входо-выходных пар в алфавите $I \times O$, получаемых при последовательных переходах из состояния a . Формально, язык $L_A^*(a)$ есть подмножество $(I \times O)^*$, и последовательность $(i_1, o_1) \dots (i_k, o_k) \in L_A^*(a)$, если и только если $o_1 \dots o_k \in out(a, i_1 \dots i_k)$. По определению, язык $L_A^*(a)$ содержит пару $(\varepsilon, \varepsilon)$. Язык $L_A^*(a_0)$ автомата

³ Как обычно, I^* обозначает множество всех последовательностей конечной длины из символов множества I

A в начальном состоянии a_0 называется языком автомата и обозначается L_A .

Вообще говоря, язык автомата описывает отображение, реализуемое этим автоматом. Последовательность $(i_1, o_1) \dots (i_k, o_k)$ принадлежит языку $L_A(a)$, если и только если автомат A в состоянии a может выдать выходную последовательность $o_1 \dots o_k$ на входную последовательность $i_1 \dots i_k$.

Если мы хотим подчеркнуть, что в автомате каждый выходной символ появляется как реакция на некоторый входной символ, то язык автомата удобно представлять в виде последовательностей входных и выходных символов. Такой язык $L_A(a)$ есть подмножество $(I \cup O)^*$, и последовательность $i_1 o_1 \dots i_k o_k$ принадлежит $L_A(a)$, если и только если $(i_1, o_1) \dots (i_k, o_k)$ принадлежит $L_A(a)$. По определению, язык $L_A(a)$ содержит пустую последовательность ϵ .

Рассмотрим регистр сдвига (рис. 1.5), состоящий из двух элементов задержки, называемых далее также *триггерами*, которые изображены в виде квадратов. Под триггером понимается автомат с двумя состояниями 0 и 1, входной и выходной алфавит которого есть $\{0, 1\}$. В начальный момент состояние триггера равно 0. Следующее состояние триггера совпадает с текущим входным символом; выходной символ равен текущему состоянию триггера. Таким образом, в начальный момент оба триггера находятся в состоянии 0.

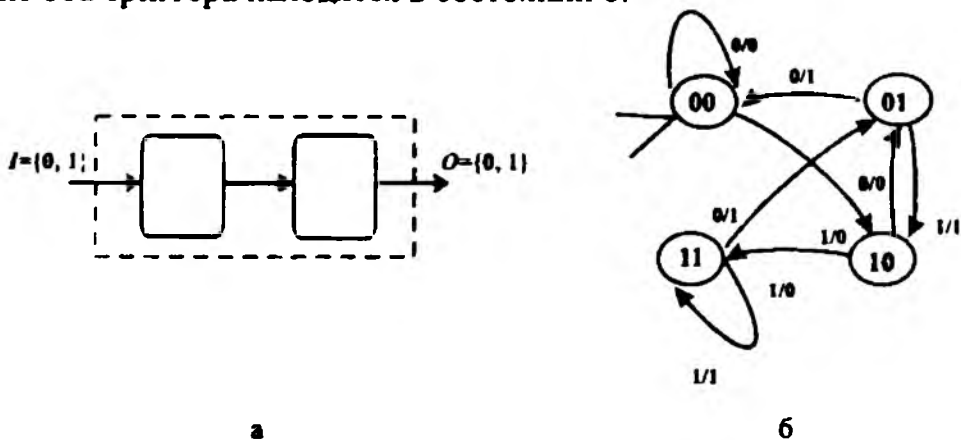


Рис. 1.5, а регистр сдвига с двумя ячейками
 б автомат, описывающий поведение регистра сдвига

При поступлении на вход регистра входного символа 0 или 1 поступивший символ записывается в первую ячейку, содержимое кото-

рой сдвигается во вторую ячейку. На выходе регистра появляется выходной символ, равный состоянию второй ячейки. Таким образом, регистр имеет 4 состояния 00, 01, 10, 11, и его поведение описывается диаграммой переходов на рис. 1.56. Непосредственной проверкой можно убедиться, что для любой входной последовательности $i_1 \dots i_k$ выходная последовательность регистра равна $00i_1 \dots i_{k-2}$. Соответственно, язык $L_A(00)$ такого регистра сдвига состоит из последовательностей $(i_1, 0) (i_2, 0) (i_3, i_1) \dots (i_k, i_{k-2})$, где $i_j \in \{0, 1\}$.

Автомат $A = (A, I, O, T_A, a_0)$ называется *связным*, если любое состояние достижимо из начального состояния. Автомат $B = (B, I, O, T_B, b_0)$ называется *подавтоматом* автомата A , если $B \subseteq A$, $b_0 = a_0$ и $T_B \subseteq T_A$.

Состояние b недетерминированного автомата $B = (B, I, O, T_B, b_0)$ называется *редукцией* состояния a недетерминированного автомата $A = (A, I, O, T_A, a_0)$ (обозначение $b \leq a$), если $L_A(b) \subseteq L_A(a)$.

Состояния b и a называются *эквивалентными* (обозначение $b \equiv a$), если b есть редукция a , и a есть редукция b . В противном случае состояния b и a не являются эквивалентными. Для частичных автоматов A и B данное определение отличается от общепринятого. В эквивалентных состояниях поведение частичных автоматов определено на одном и том же множестве входных последовательностей. Непосредственной проверкой можно убедиться, что регистр сдвига не имеет эквивалентных состояний. Такие автоматы часто называют *минимальными* или *приведенными*.⁴

Автомат $B = (B, I, O, T_B, b_0)$ есть *редукция* автомата $A = (A, I, O, T_A, a_0)$, если $L_B \subseteq L_A$. Если $L_B = L_A$, то автоматы A и B называются *эквивалентными*. Для каждого недетерминированного автомата существует эквивалентный *наблюдаемый* автомат. Для детерминированных полностью определенных автоматов отношения редукции и эквивалентности совпадают.

В качестве примера опишем автоматом поведение логической схемы, приведенной на рис. 1.6, которая используется для умножения многочлена $a(x) = a_k x^k + a_{k-1} x^{k-1} + \dots + a_1 x + a_0$ на фиксированный многочлен $h(x) = x^2 + 1$.⁵ Такое умножение, в частности, необходимо при помехоустойчивом кодировании с использованием циклических кодов.

⁴ Более подробно см. ч. I настоящего пособия

⁵ В данном примере знак «+» обозначает сложение по модулю 2.

Вектор коэффициентов многочлена $a(x)$ есть информационное слово длины $(k + 1)$. Многочлен $g(x) = g_2x^2 + g_1x + g_0$ является многочленом, порождающим циклический код, и в нашем случае есть многочлен $x^2 + 1$, т.е. $g_2 = g_0 = 1$ и $g_1 = 0$. На выходе схемы получается кодовое слово длины $(k + 3)$. Предполагается, что первоначально все элементы задержки содержат нули, а на вход схемы поступают коэффициенты многочлена $a(x)$, начиная с коэффициентов высших порядков, после чего следуют два нуля.

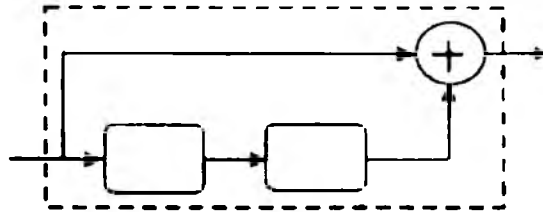


Рис 1.6. Логическая схема для умножения на многочлен

Непосредственной проверкой можно убедиться, что произведение равно

$$a(x)g(x) = a_k g_2 x^{k+2} + (a_{k-1} g_2 + a_k g_1) x^{k+1} + \dots + (a_0 g_1 + a_1 g_0) x + a_0 g_0 = a_k x^{k+2} + a_{k-1} x^{k+1} + \dots + a_1 x + a_0.$$

Когда на вход схемы подается первый коэффициент a_k многочлена $a(x)$, то на выходе появляется первый коэффициент произведения $a(x)g(x)$, равный $a_k = a_k g_2$, и содержимое всех ячеек сдвигается на одну ячейку вправо. В следующий такт времени на входе схемы появляется следующий коэффициент a_{k-1} . Соответственно (рис. 1.6), выход схемы в данном такте будет равен a_{k-1} , т.е. выход равен величине второго коэффициента в произведении $a(x)g(x)$. Аналогично в следующем такте на входе появится a_{k-2} , а ячейки регистра сдвига будут содержать элементы a_{k-1}, a_k . Выход схемы равен $(a_{k-2} + a_k)$, то есть величине третьего коэффициента произведения $a(x)g(x)$. Дальнейшие операции производятся аналогичным образом. После $[(k + 3) - 1]$ сдвигов в ячейках регистра содержатся элементы a_0, a_1 , и при входном символе 0 выход схемы равен $a_1 = a_0 g_1 + a_1 g_0$, т.е. предпоследнему коэффициенту произведения $a(x)g(x)$. После $(k + 3)$ сдвигов ячейки регистра содержат элементы 0, 0, и выход равен $a_0 = a_0 g_0$ — последнему коэффициенту произведения $a(x)g(x)$, так что произведение получено полностью.

На рис. 1.7 представлена диаграмма переходов автомата, описывающего поведение логической схемы на рис. 1.6.

Пусть информационное слово, которое нужно закодировать, равно 1101; соответствующее кодовое слово получится на выходе автомата, если на вход автомата подана входная последовательность 110100. Найдем выходную последовательность автомата при подаче на его вход последовательности 110100 в начальном состоянии 00. Согласно диаграмме переходов выходная последовательность есть 111001, и после выдачи этой последовательности автомат вернется в начальное состояние 00. Таким образом, информационное слово 1101, соответствующее многочлену $x^3 + x^2 + 1$, будет переведено автоматом в кодовое слово 111001, соответствующее многочлену $x^5 + x^4 + x^3 + 1$. Непосредственной проверкой можно убедиться, что $x^5 + x^4 + x^3 + 1 = (x^3 + x^2 + 1)(x^2 + 1)$.

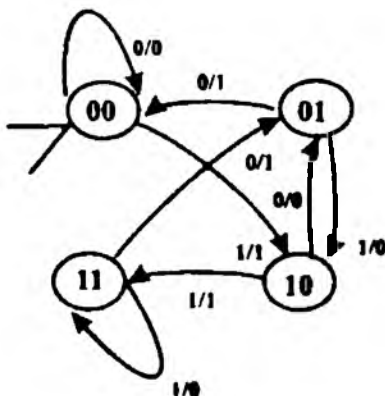


Рис. 1.7. Диаграмма переходов автомата, описывающего поведение логической схемы для умножения на многочлен

Задачи для самостоятельного решения

1. Для автомата по продаже билетов (рис. 1.36) предусмотреть возможность выдачи сдачи, если сумма опущенных в автомат монет превышает стоимость билета. Предусмотреть сигнал сброса, если клиент передумал покупать билет и хочет получить обратно деньги.

2. Является ли автомат на рис. 1.7 полностью определенным? Детерминированным? Имеет ли этот автомат эквивалентные состояния? Почему?

3. Являются ли регистры сдвига с k и $(k + 1)$ ячейками, $k \geq 1$, эквивалентными? Почему?

4. Описать автоматом поведение инвертора, конъюнктора и дизъюнктора на два входа (рис. 1.8).

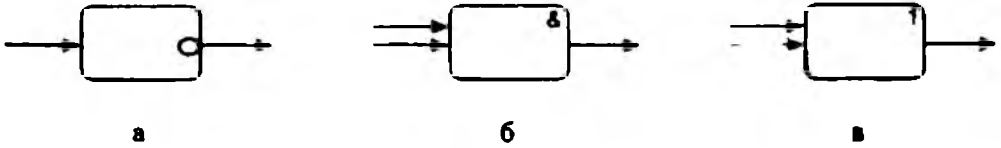


Рис 1.8, а инвертор
 б конъюнктор на два входа
 в дизъюнктор на два входа

5°. Каким образом можно описать поведение официанта (рис. 1.4в) с помощью детерминированного автомата, предположив, к примеру, что покупатель может получить не более двух чашек кофе без предварительной оплаты.

1.3. Представление автоматов полуавтоматами

Язык любого автомата является регулярным и, соответственно, может быть представлен с помощью полуавтомата. *Полуавтоматом* называется пятерка $P = (P, J, T_P, p_0, F_P)$, где P – конечное множество состояний с выделенным начальным состоянием p_0 , J – алфавит, $T_P \subseteq J \times P \times P$ – отношение переходов и F_P – множество финальных состояний. Полуавтомат называется *детерминированным*, если для любой пары $(p, j) \in P \times J$ существует не более одного состояния $p' \in P$ такого, что $(p, j, p') \in T_P$. В противном случае полуавтомат называется *недетерминированным*.

Языком полуавтомата P в состоянии p называется множество последовательностей, которые переводят полуавтомат из состояния p в одно из финальных состояний. По определению, пустое слово принадлежит языку полуавтомата в состоянии p , если и только если p принадлежит множеству финальных состояний. Два полуавтомата *эквивалентны*, если их языки совпадают. Известно, что для любого полуавтомата $P = (P, J, T_P, p_0, F_P)$ можно построить эквивалентный ему детерминированный полуавтомат $D_P = (D, J, T_D, d_0, F_D)$.

Для представления языка автомата $A = (A, I, O, T_A, a_0)$ можно построить полуавтоматы двумя способами. В первом случае строится полуавтомат $P_A^* = (A, I \times O, T_{P_A^*}, a_0, A)$. Полуавтомат P_A^* имеет то же самое множество состояний и начальное состояние, что и автомат A , каждое состояние является финальным. Множество действий полуавтомата есть декартово произведение входного и выходного алфавитов автомата, тройка $(a, (i, o), a') \in T_{P_A^*}$, если и только если $(a, i, o, a') \in T_A$.

Язык полуавтомата P_A^* совпадает с языком L_A^* автомата A .

Во втором случае строится полуавтомат $P_A^\cup = (A \cup (A \times I), I \cup O, T_{P_A}, a_0, A)$. Множество состояний полуавтомата P_A^\cup есть объединение множеств $A \cup (A \times I)$; финальными являются состояния из множества A . Для каждого состояния $a \in A$ и входного символа $i \in I$ множество T_{P_A} содержит переход $(a, i, (a', i))$, если и только если $\exists o \in O ((a, i, o, a') \in T_A)$. Для каждого состояния $(a, i) \in A \times I$ и выходного символа $o \in O$ множество T_{P_A} содержит переход $((a, i), o, a')$, если и только если $(a, i, o, a') \in T_A$. Язык полуавтомата P_A^\cup совпадает с языком L_A автомата A .

Полуавтоматы P_A^* и P_A^\cup детерминированные, если и только если автомат A наблюдаемый.

Автоматы A и B эквивалентны, если и только если эквивалентны полуавтоматы P_A^* и P_B^* (полуавтоматы P_A^\cup и P_B^\cup).

Пример. Построим полуавтоматы P^* и P^\cup для автомата, описывающего поведение официанта в кофейне (рис. 1.4в). Множество состояний полуавтомата P^* совпадает с множеством состояний a, b, c автомата, описывающего поведение официанта; множество действий полуавтомата P^* есть $(M, Th), (M, S), (M, C), (M, B), (M, Es), (Ep, Th), (Ep, S), (Ep, C), (Ep, B), (Ep, Es), (E, Th), (E, S), (E, C), (E, B), (E, Es), (L, Th), (L, S), (L, C), (L, B), (L, Es)$ (рис. 1.9а). Состояниями полуавтомата P^\cup являются состояния a, b, c автомата и все пары $(a, Ep), (a, E), (a, M), (a, L), (b, Ep), (b, E), (b, M), (b, L), (c, Ep), (c, E), (c, M), (c, L)$, множество действий полуавтомата есть множество $M, Ep, E, L, Th, S, C, B, Es$. Часть полуавтомата P^\cup , содержащая все состояния, достижимые из начального состояния, приведена на рис. 1.9б⁶.

⁶ Нефинальные состояния полуавтомата заштрихованы

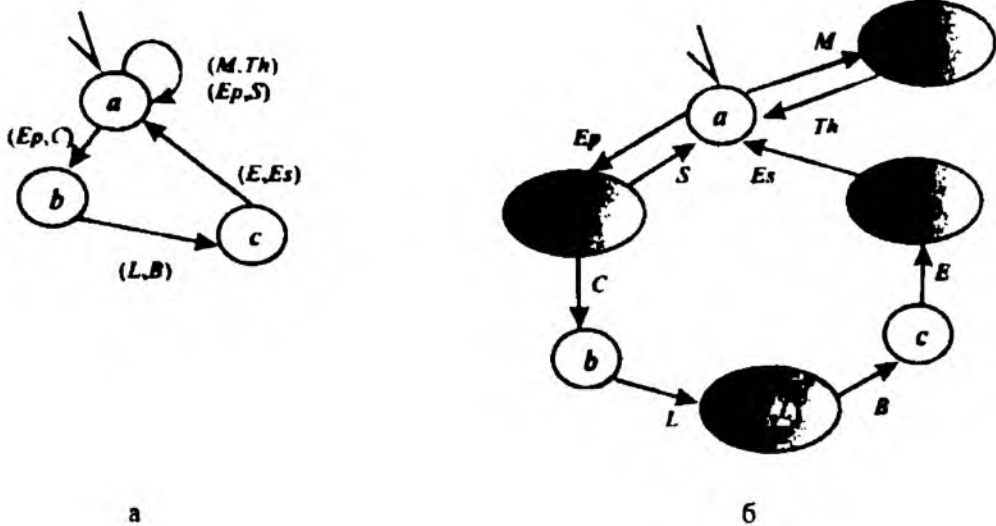


Рис. 1.9, а полуавтомат P^a
 б полуавтомат P^b

Задачи для самостоятельного решения

1. Представить язык автомата на рис. 1.3б двумя различными полуавтоматами.

2. Описать двумя различными полуавтоматами регистр сдвига с двумя элементами задержки, одним входом и одним выходом (рис. 1.5), при условии, что сигнал сброса переводит обе ячейки регистра в состояние 0. Являются ли построенные полуавтоматы детерминированными? Почему?

1.4. Композиция автоматов

Важным этапом при иерархическом синтезе дискретных систем является представление системы в виде композиции других, более простых в некотором смысле подсистем. Для формализации этого процесса необходимо ввести операции композиции. В настоящем пособии мы предполагаем, что поведение всех подсистем и системы в целом описывается конечными автоматами, и подсистемы взаимодействуют, обмениваясь словами в их языках. Рассматриваются два вида композиции: синхронная, когда действия всех компонент синхронизированы, и параллельная, когда компоненты работают по мере их готовности. Синхронная композиция обычно используется при синтезе и анализе цифровых схем; параллельная композиция востребована при описании совместного поведения различных процессов, в частности, при описа-

нии совместного поведения телекоммуникационных протоколов. Для простоты представления мы вводим только композицию двух автоматов, однако все результаты справедливы и для более сложных многокомпонентных композиций. Необходимые ссылки интересующийся читатель может найти в списке дополнительной литературы.

1.4.1. Синхронная композиция конечных автоматов

Синхронное функционирование многокомпонентной системы предполагает наличие глобального синхросигнала, позволяющего всем компонентам системы по этому сигналу принять соответствующий входной сигнал, переключиться из текущего состояния в следующее и произвести выходной сигнал. При подаче внешнего входного сигнала система вырабатывает согласованный набор внутренних сигналов, переходит в следующее состояние и выдает внешний выходной сигнал. Для формального определения синхронной композиции автоматов напомним операции проекции и расширения языков⁷.

Пусть L есть язык в алфавите $I \times V$. Рассмотрим язык $L_{\downarrow I}$, в котором в каждой паре (i, v) каждого слова из языка L удаляется символ $v \in V$. Язык $L_{\downarrow I} \subseteq I^*$ называется *проекцией* языка L на алфавит I или просто *I -проекцией* языка L . Проекция $L_{\downarrow I}$ показывает, какие последовательности в алфавите I участвуют в образовании слов языка L . Если язык L регулярный и представлен полуавтоматом P , то полуавтомат $P_{\downarrow I}$ представляющий язык $L_{\downarrow I}$ получается из P «стиранием» пометок $v \in V$ на переходах. Полуавтомат, полученный после «стирания», может оказаться недетерминированным. Если необходимо, то полуавтомат можно детерминизировать.

Пусть L есть язык в алфавите I . Рассмотрим язык $L_{\uparrow V}$ в котором каждый символ $i \in I$ в каждом слове заменяется всеми парами (i, v) , $v \in V$. Язык $L_{\uparrow V}$ называется *расширением* языка L на алфавит V или просто *V -расширением* языка L . Расширение $L_{\uparrow V}$ показывает, какие последовательности в алфавите $I \times V$ можно получить с помощью слов языка L . Если язык L регулярный и представлен полуавтоматом P , то полуавтомат $P_{\uparrow V}$ представляющий язык $L_{\uparrow V}$ получается из P добавлением на каждом переходе, помеченном символом i , всех пар (i, v) ,

⁷ Подробно см. в первой части пособия

$v \in V$. Если полуавтомат \mathcal{P} детерминированный, то полуавтомат $\mathcal{P}_{\uparrow V}$ также детерминированный. Если язык L состоит из одного пустого слова ε , то язык $L_{\uparrow V}$ также состоит из одного пустого слова ε .

Рассмотрим композицию автоматов на рис. 1.10, в которой

- автомат A имеет входной алфавит $I_1 \times V$, выходной алфавит $O_1 \times U$ и отношение переходов T_A ;
- автомат B имеет входной алфавит $I_2 \times U$, выходной алфавит $O_2 \times V$ и отношение переходов T_B .

Синхронная композиция автоматов A и B имеет

- входной алфавит $I = I_1 \times I_2$;
- выходной алфавит $O = O_1 \times O_2$;
- под действием входного символа $(i_1, i_2) \in I_1 \times I_2$ композиция «вырабатывает» выходной символ $(o_1, o_2) \in O_1 \times O_2$ и пару соответствующих внутренних символов $(v, u) \in V \times U$. Под парой соответствующих внутренних символов $(v, u) \in V \times U$ в текущих состояниях a и b мы понимаем, что

$\exists a' \in A \exists b' \in B [(a, (i_1, v), (o_1, u), a') \in T_A \& (b, (i_2, u), (o_2, v), b') \in T_B]$.

Таким образом, при подаче входных символов $i_1 \in I_1$ и $i_2 \in I_2$ компоненты A и B в текущих состояниях вырабатывают пару внутренних выходных символов $u \in U$ и $v \in V$ таких, что u есть внутренний выходной сигнал компоненты A на входной сигнал (i_1, v) и, соответственно, v есть внутренний выходной сигнал компоненты B на входной сигнал (i_2, u) . Кроме того, автомат A переходит в следующее состояние a' под действием входного символа (i_1, v) , а автомат B переходит в следующее состояние b' под действием входного символа (i_2, u) .

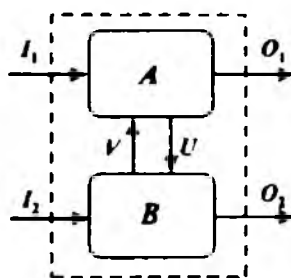


Рис. 1.10 Композиция автоматов A и B

При таком функционировании последовательность $\alpha \in (I_1 \times I_2 \times O_1 \times O_2)^*$ принадлежит синхронной композиции автоматов A и B , если и только если α принадлежит $(I_1 \times I_2 \times O_1 \times O_2)$ -

проекции пересечения $(I_2 \times O_2)$ -расширения языка L_A^* автомата A и $(I_1 \times O_1)$ -расширения языка L_B^* автомата B (более подробно см. часть 1 настоящего пособия). Расширения языков L_A^* и L_B^* необходимы, чтобы описать слова в композиции, в образовании которых может участвовать компонента A (B). После расширения языки компонент определены в одном алфавите $I_1 \times I_2 \times O_1 \times O_2 \times V \times U$. Формально, поведение синхронной композиции можно описать следующим автоматом.

Синхронной композицией автоматов A и B называется приведенный наблюдаемый автомат $A \bullet B$ с языком $L_A^* \bullet L_B^* = [(L_A^*) \uparrow_{I_2 \times O_2} \cap (L_B^*) \uparrow_{I_1 \times O_1}] \downarrow_{I_1 \times I_2 \times O_1 \times O_2}$ ⁸.

Известно, что, если L_A^* и L_B^* суть языки, реализуемые некоторыми конечными автоматами, то язык $L_A^* \bullet L_B^*$ также реализуется некоторым, возможно, частичным и недетерминированным автоматом с входным алфавитом $I_1 \times I_2$ и выходным алфавитом $O_1 \times O_2$.

В случае синхронной композиции, вообще говоря, не требуется, чтобы присутствовали все алфавиты I_1, I_2, O_1, O_2, V, U . При отсутствии некоторых алфавитов получаются более простые топологии композиции. Например, на рис. 1.11 приведена композиция автоматов A и B без обратных связей.

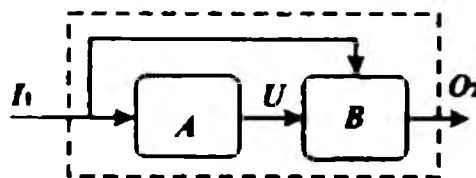


Рис. 1.11 Каскадная (без обратных связей) композиция автоматов A и B

Пример. Рассмотрим логическую сеть из двух автоматов: элемента задержки и комбинационной схемы, реализующей сложение по модулю 2 (рис. 1.12а).

⁸ В качестве синхронной композиции, вообще говоря, можно рассматривать любой автомат с языком $[(L_A^*) \uparrow_{I_2 \times O_2} \cap (L_B^*) \uparrow_{I_1 \times O_1}] \downarrow_{I_1 \times I_2 \times O_1 \times O_2}$, т.к. все эти автоматы эквивалентны

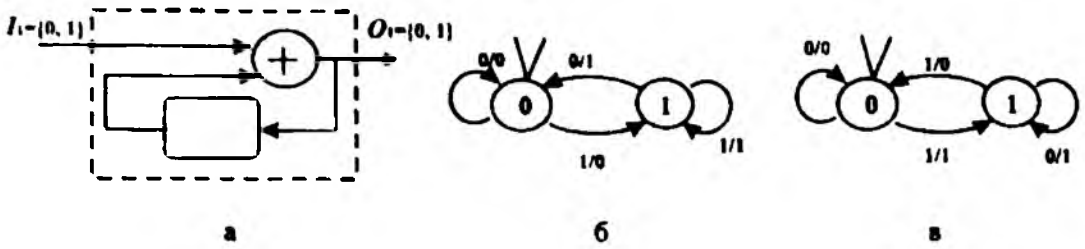


Рис. 1.12. а логическая сеть из элемента задержки и сумматора по модулю 2
 б автоматное описание элемента задержки
 в синхронная композиция элемента задержки и сумматора

Используя оператор синхронной композиции, получаем автомат с диаграммой переходов на рис. 1.12в.

Автомат «считает» единицы в подаваемой входной последовательности. При наличии нечетного числа единиц во входной последовательности на выходе автомата появляется 1; при наличии четного числа единиц на выходе автомата появляется 0.

Введенное выше определение синхронной композиции не накладывает никаких ограничений на автоматы-компоненты и справедливо, в частности, для частичных (в том числе недетерминированных) автоматов. Однако при использовании этого определения для таких автоматов в различных приложениях желательно проверить, соответствует ли такое определение физическому смыслу композиции.

В большинстве приложений обычно предполагается, что автоматы *A* и *B* полностью определенные и детерминированные, и накладываются дополнительные условия, для того, чтобы композиция автоматов *A* и *B* была также полностью определенным и детерминированным автоматом. Для каскадной композиции на рис. 1.11 последнее всегда выполняется. Для композиции с обратной связью, например, для композиции на рис. 1.10, обычно один из автоматов *A* или *B* выбирается автоматом Мура, т.е. выходной символ этого автомата зависит только от его текущего состояния.

Пример. Рассмотрим композицию (рис. 1.13а) элемента задержки (триггера) и комбинационной схемы, реализующей отображение, представленное диаграммой на рис. 1.13б. Комбинационная схема имеет два входа и два выхода. Один из входов комбинационной схемы является внешним (первый символ в четверке iv/uo); второй входной символ комбинационной схемы «снимается» с выхода триггера. Первый выходной символ в четверке iv/uo «подается» на вход триггера,

второй выходной символ «снимается» с внешнего выхода комбинационной схемы.

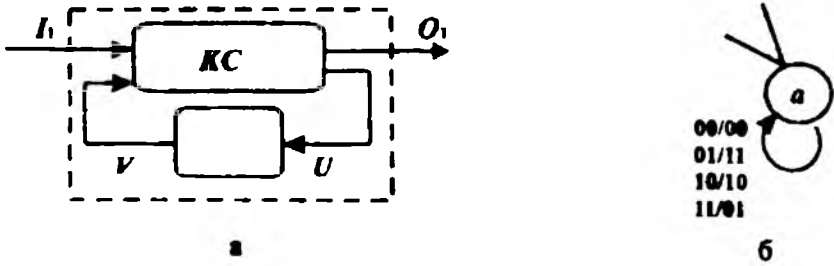


Рис. 1.13. а логическая сеть из триггера и комбинационной схемы КС
б диаграмма переходов комбинационной схемы КС

Язык комбинационной схемы расширять не требуется. Полуавтомат, представляющий расширение языка элемента задержки на внешние алфавиты, представлен на рис. 1.14 (в алфавите $I_1 \times V \times U \times O_1$).

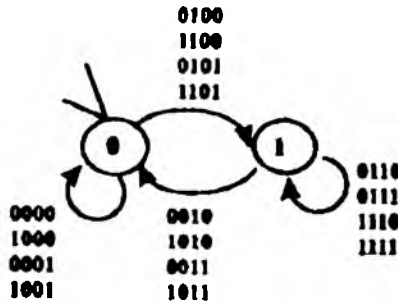


Рис. 1.14. Расширение языка элемента задержки на внешние алфавиты

Автомат на рис. 1.15 есть синхронная композиция автоматов на рис. 1.126 и рис. 1.136.

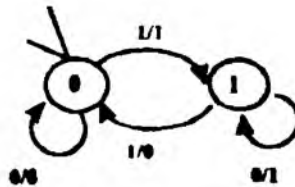


Рис. 1.15. Синхронная композиция автоматов-компонент (рис. 1.126 и 1.136)

1.4.2. Параллельная композиция конечных автоматов

Асинхронное функционирование многокомпонентной системы предполагает отсутствие глобального синхросигнала, позволяющего всем компонентам системы по этому сигналу переключиться из теку-

шего состояния в следующее и произвести выходной символ. В настоящем пособии мы рассматриваем только один из способов асинхронного функционирования, когда компоненты взаимодействуют в режиме диалога, т.е. в каждый момент времени активной является только одна компонента. Такое функционирование можно описать с использованием операции параллельной композиции автоматов-компонент. При подаче внешнего входного сигнала на одну из компонент (рис. 1.10) компонента вырабатывает внешний или внутренний выходной сигнал. Если компонента вырабатывает внутренний сигнал, то компоненты работают в режиме диалога, пока компоненты не перестанут обмениваться внутренними сигналами, и одна из них не выдаст внешний выходной сигнал. После этого композиция готова обработать следующий внешний входной сигнал. Для формального определения параллельной композиции автоматов напомним операции ограничения и распространения языков (подробно см. в первой части пособия).

Пусть L есть язык в алфавите $I \cup V$. Рассмотрим язык $L_{\cup I}$, в котором каждый символ $v \in I$ заменяется пустым словом ε . Язык $L_{\cup I}$ называется *ограничением языка L на алфавит I* или просто *I -ограничением языка L* . Ограничение $L_{\cup I}$ показывает, какие последовательности в алфавите I являются частями слов языка L . Если язык L регулярный и представлен полуавтоматом P , то полуавтомат $P_{\cup I}$, представляющий язык $L_{\cup I}$, получается из P «стиранием» пометок $v \in I$. В полуавтомате, полученном после «стирания», можно потребовать удаления переходов, помеченных пустым словом. После этого полуавтомат может оказаться недетерминированным. Если необходимо, то полуавтомат можно детерминизировать.

Пусть L есть язык в алфавите I . Рассмотрим язык $L_{\cap V}$, в котором в каждом слове после каждого начального отрезка слова вставляются все конечные последовательности из алфавита V . Язык $L_{\cap V}$ называется *распространением языка L на алфавит V* или просто *V -распространением языка L* . Распространение $L_{\cap V}$ показывает, какие последовательности в алфавите $I \cup V$ можно получить с помощью слов языка L . Если язык L регулярный и представлен полуавтоматом P , то полуавтомат $P_{\cap V}$, представляющий язык $L_{\cap V}$, получается из P добавлением в каждом состоянии петли, помеченной всеми символами из алфавита V . Если полуавтомат P детерминированный, то полуавтомат

$P_{\cap V}$ также детерминированный. Если язык L состоит из одного пустого слова ε , то язык $L_{\cap V}$ состоит из всех слов в алфавите \mathcal{W} .

Рассмотрим композицию автоматов на рис. 1.10, в которой

- автомат A имеет входной алфавит $I_1 \cup V$, выходной алфавит $O_1 \cup U$ и отношение переходов T_A ;

- автомат B имеет входной алфавит $I_2 \cup U$, выходной алфавит $O_2 \cup V$ и отношение переходов T_B ;

- алфавиты I_1, I_2, O_1, O_2, V и U попарно не пересекаются.

Параллельная композиция автоматов A и B имеет

- входной алфавит $I = I_1 \cup I_2$;

- выходной алфавит $O = O_1 \cup O_2$;

- под действием входного символа $i, i \in I_1 \cup I_2$ композиция «вырабатывает» внешний выходной символ $o \in O_1 \cup O_2$ или внутренний выходной символ, который является входным для другой компоненты;

- следующий входной символ может быть подан на композицию только после того, как компоненты закончили внутренний диалог; если компоненты суть детерминированные полностью определенные автоматы, то это означает, что композиция отреагировала внешним выходным сигналом на предыдущий внешний входной сигнал.

При таком функционировании последовательность $\alpha \in [(I_1 \cup I_2)(O_1 \cup O_2)]^*$ принадлежит параллельной композиции автоматов A и B , если и только если α принадлежит $(I_1 \cup I_2 \cup O_1 \cup O_2)$ -ограничению пересечения $(I_2 \cup O_2)$ -распространения языка L_A автомата A и $(I_1 \cup O_1)$ -распространения языка L_B автомата B (более подробно см. часть I настоящего пособия). Расширения языков L_A и L_B необходимы, чтобы языки были определены в одном алфавите $I_1 \cup I_2 \cup O_1 \cup O_2 \cup V \cup U$. Кроме того, чтобы «запретить» подачу следующего входного символа до выдачи выходного символа на предыдущий входной символ, мы должны пересечь полученный язык с $(IO)^*$. Формально поведение параллельной композиции можно описать следующим автоматом.

Параллельной композицией автоматов A и B называется приведенный наблюдаемый автомат $A \diamond B$ с языком $(L_A \diamond L_B) \cap (IO)^*$, где $L_A \diamond L_B = [(L_A) \uparrow \text{мл} \downarrow \alpha_1 \cap (L_B) \uparrow \text{мл} \downarrow \alpha_2] \uparrow \text{мл} \downarrow \alpha_1 \downarrow \alpha_2$.⁹

Известно, что если L_A и L_B суть языки, реализуемые некоторыми конечными автоматами, то $(L_A \diamond L_B) \cap (IO)^*$ также реализуется некоторым, возможно, частичным автоматом с входным алфавитом $I_1 \cup I_2$ и выходным алфавитом $O_1 \cup O_2$. Подобно синхронной композиции, некоторые из алфавитов могут быть пустыми, и в этом случае говорят о различных топологиях композиции.

Подобно определению синхронной композиции, определение параллельной композиции справедливо и для частичных, в том числе, недетерминированных, автоматов. Однако подобно операции синхронной композиции, необходимо убедиться, что введенное нами определение соответствует физическому смыслу композиции.

Заметим, что параллельная композиция полностью определенных и детерминированных автоматов A и B может оказаться частичным автоматом. Причиной является тот факт, что после подачи некоторой входной последовательности компоненты начинают вести «бесконечный» диалог, не выдавая никаких внешних сигналов. Для композиции на рис. 1.10, вообще говоря, не известно, какими свойствами должны обладать полностью определенные детерминированные автоматы-компоненты, чтобы в композиции не было осцилляции.

Пример. На рис. 1.16 представлена сеть из двух автоматов, в которой автомат B не имеет внешних входных и выходных символов.

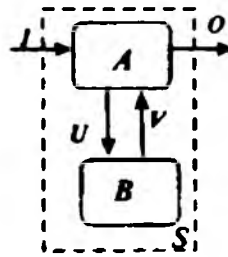


Рис. 1.16. Автоматная сеть

⁹ Подобно синхронной композиции, в качестве параллельной композиции можно рассматривать любой автомат с языком $[(L_A) \uparrow \text{мл} \downarrow \alpha_1 \cap (L_B) \uparrow \text{мл} \downarrow \alpha_2] \uparrow \text{мл} \downarrow \alpha_1 \downarrow \alpha_2$, т.к. все эти автоматы эквивалентны

Для этой сети построим параллельную композицию автоматов, которые описывают поведение официанта и кофейной машины, т.е. построим автомат, описывающий функционирование кофейни (рис. 1.4а). На рис. 1.17а,б показаны соответствующие полуавтоматы. Рис. 1.17в иллюстрирует необходимое расширение полуавтомата кофейной машины. Пересечение полуавтоматов и ограничение пересечения на внешние алфавиты показаны, соответственно, на рис. 1.17г и 1.17д.

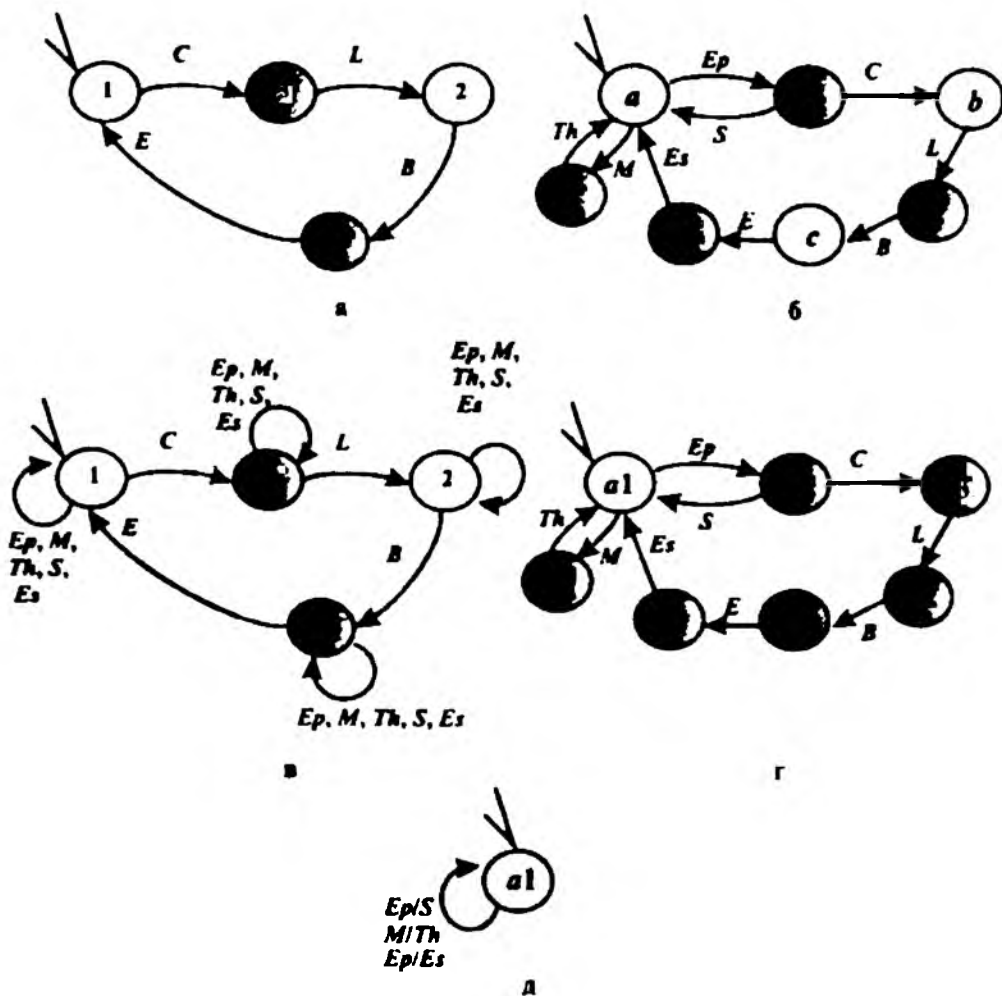


Рис. 1.17, а полуавтомат, описывающий поведение официанта
 б полуавтомат, описывающий поведение кофейной машины
 в расширение полуавтомата, описывающего поведение кофейной машины
 г пересечение полуавтоматов
 д ограничение пересечения на внешние алфавиты

Задачи для самостоятельного решения

1. Построить синхронную композицию двух последовательно соединенных комбинационных схем (рис. 1.18). Указание: входы и выходы с одинаковыми алфавитами считаются отождествленными.

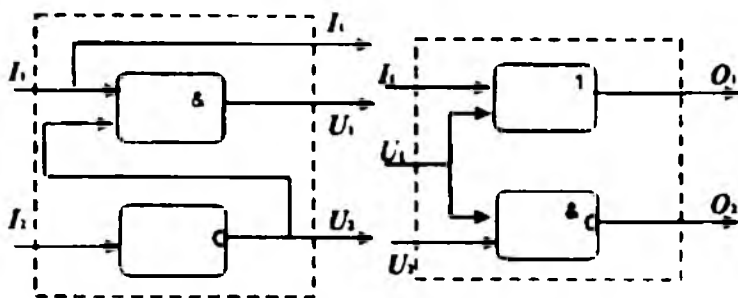


Рис. 1.18. Две комбинационные схемы

2. Построить последовательную композицию двух регистров сдвига, каждый из которых состоит из двух элементов задержки.

1.5. Автоматные неравенства и уравнения

В настоящем пособии мы ограничимся введением автоматного уравнения для композиции на рис. 1.10. Более общие автоматные уравнения заинтересованный читатель может найти в литературных источниках, которые приведены в списке дополнительной литературы.

Пусть заданы полностью определенные наблюдаемые автоматы $S = (S, I_1 \times I_2, O_1 \times O_2, T_S, s_0)$ и $A = (A, I_1 \times V, O_1 \times U, T_A, a_0)$ (рис. 1.10). Выражение $A \bullet X \equiv S$, в котором X есть автомат с входным алфавитом $I_2 \times U$ и выходным алфавитом $O_2 \times V$, называется *синхронным автоматным уравнением*. Автоматное уравнение можно рассматривать как автоматный аналог деления целых чисел. В случае целых чисел, когда найти целое частное не удастся, обычно находится наибольшее целое число такое, что произведение делителя и частного не превосходит делимое. Автоматным аналогом такой ситуации является решение автоматного неравенства, т.е. выражение $A \bullet X \leq S$ называется *синхронным автоматным неравенством*. Автомат B с входным алфавитом $I_2 \times U$ и выходным алфавитом $O_2 \times V$ называется *решением неравенства* $A \bullet X \leq S$, если $A \bullet B \leq S$. Автомат B называется *решением уравнения* $A \bullet X \equiv S$, если $A \bullet B \equiv S$.

Решение *Largest* называется *наибольшим решением* неравенства $A \bullet X \leq S$ (уравнения $A \bullet X \cong S$), если каждое решение неравенства (уравнения) есть редукция автомата *Largest*.

Введенное таким образом автоматное неравенство (уравнение) передает физический смысл понятия его решения. Решение автоматного уравнения $A \bullet X \cong S$ соответствует автомату, который в композиции с автоматом *A* определяет поведение, описанное автоматом *S*.

Пример. Пусть *T* – элемент задержки (рис. 1.12б) и *S* – автомат на рис. 1.15. Тогда автомат *K* на рис. 1.13б будет решением уравнения $T \bullet X \cong S$ для логической сети на рис. 1.13а, т.к. $T \bullet K \cong S$. Однако уравнение может иметь и другие решения, в частности, всякий автомат, эквивалентный *K*, с любым числом состояний будет тоже решением уравнения.

Автоматное уравнение может не иметь решений. Например, если в уравнении $A \bullet X \cong S$ автомат *A* в начальном состоянии на любой входной символ (i_1, v) выдает выходной символ $o_1 = 0$, в то время как автомат *S* в начальном состоянии на любой входной символ (i_1, i_2) выдает выходной символ $o_1 = 1$, то никакой автомат *B* в композиции с автоматом *A* не может быть эквивалентным *S*, т.е. в этом случае уравнение $A \bullet X \cong S$ не имеет решения.

С другой стороны, в ряде случаев уравнение $A \bullet X \cong S$ может иметь не одно, а множество решений. Предположим, что автоматы *B* и *S* для композиции на рис. 1.11 выдают выходной символ $o_2 = 0$ на любой входной символ в любом состоянии. В этом случае любой автомат *A* в композиции с автоматом *B* будет эквивалентен автомату *S*, т.е. любой автомат *A* будет решением уравнения $X \bullet B \cong S$.

Пусть заданы полностью определенные наблюдаемые автоматы $S = (S, I_1 \cup I_2, O_1 \cup O_2, T_S, s_0)$ и $A = (A, I_1 \cup V, O_1 \cup U, T_A, a_0)$ (рис. 1.10). Выражение $A \diamond X \cong S$, в котором *X* есть автомат с входным алфавитом $I_2 \cup U$ и выходным алфавитом $O_2 \cup V$, называется *параллельным автоматным уравнением*. Соответственно, выражение $A \diamond X \leq S$ называется *параллельным автоматным неравенством*. Автомат *B* с входным алфавитом $I_2 \cup U$ и выходным алфавитом $O_2 \cup V$ называется *решением неравенства* $A \diamond X \leq S$, если $A \diamond B \leq S$. Автомат *B* называется *решением уравнения* $A \diamond X \cong S$, если $A \diamond B \cong S$.

Решение *Largest* называется *наибольшим решением* неравенства $A \diamond X \leq S$ (уравнения $A \diamond X \cong S$), если каждое решение неравенства (уравнения) есть редукция автомата *Largest*.

Введенное таким образом автоматное неравенство (уравнение) передает физический смысл его решения. Решение автоматного уравнения $A \diamond X \cong S$ соответствует автомату, который при параллельной композиции с автоматом A , определяет поведение, описанное автоматом S .

Мы далее решаем автоматные уравнения и неравенства только для полностью определенных автоматов A и S . Методы решения уравнений для частичных автоматов A и S заинтересованный читатель может найти в списке дополнительной литературы.

Задачи для самостоятельного решения

1. Проверить, является ли кофейная машина с одним состоянием (рис. 1.19а) решением уравнения $Waiter \diamond X \cong Coffee-shop$, где $Waiter$ – автомат на рис. 1.4в, а $Coffee-shop$ – автомат на рис. 1.26.

2. Проверить, является ли комбинационная схема (рис. 1.19б) решением уравнения $T \bullet X \cong S$, где T – триггер (рис. 1.12б), S – автомат на рис. 1.12в.



Рис 1.19 а кофейная машина с одним состоянием
б комбинационная схема

1.6. Решение автоматных уравнений в различных приложениях

В этом разделе мы представляем ряд задач, которые связаны с анализом или синтезом компоненты автоматной сети, при условии, что поведение всей сети и остальных компонент известно. Проблема формализуется как решение уравнения $A @ X \approx S$, где A описывает поведение известной части системы, S описывает требуемое поведение всей системы, $@$ – оператор синхронной или параллельной композиции и \approx – отношение редукции или эквивалентности. В качестве примеров можно привести оптимизацию цифровых схем, синтез контроллеров, задачи тестирования и диагностики, формирование выигрыш-

ной стратегии в теории игр, автоматные криптосистемы и так далее. Рассмотрим подробнее, каким образом задачи оптимизации цифровых схем, синтеза контроллеров, синтеза дискретных систем, тестирования и диагностики могут быть решены с использованием автоматных уравнений.

1.6.1. Оптимизация цифровых схем

Исторически автоматное уравнение впервые было использовано для решения задачи оптимизации цифровых схем. В качестве примера рассматривалась последовательная сеть из двух автоматов (рис. 1.20), и было показано, каким образом можно оптимизировать хвостовую компоненту B .

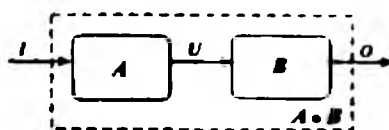


Рис. 1.20. Последовательная сеть из двух автоматов

Известно, что существует множество автоматов, которые могут заменить компоненту B без изменения внешнего функционирования сети. Все такие автоматы суть решения автоматного уравнения $A \bullet X \equiv A \bullet B$. Таким образом, найдя общее решение такого уравнения, мы сможем выбрать оптимальную (например, по числу состояний) хвостовую компоненту. В следующей главе мы покажем, что если A и B – полностью определенные детерминированные автоматы, то общее решение такого автоматного уравнения можно представить в виде частичного автомата.

Рассмотрим функционирование композиции на рис. 1.20. На вход композиции поступает внешний входной символ $i \in I$, компонента A в ответ на символ i вырабатывает выходной символ $u \in U$, который поступает на вход компоненты B . Компонента B преобразует символ u во внешний выходной символ $o \in O$. Соответственно, поведение компоненты B существенно для внешнего поведения только на тех входных последовательностях, которые могут появиться на выходе головной компоненты A . На всех остальных входных последовательностях поведение компоненты B может быть любым. Таким образом, можно построить частичный автомат B' , поведение которого определено только на выходных последовательностях автомата A ; причем на таких

последовательностях поведение автомата B совпадает с поведением автомата B . Любой полностью определенный автомат D , квазиэквивалентный автомату B , является решением уравнения и может заменить компоненту B без изменения внешнего поведения композиции, т.е. для любого полностью определенного автомата D , квазиэквивалентного автомату B , справедливо $A \bullet D \equiv A \bullet B$.

Пример. Рассмотрим последовательную композицию двух комбинационных схем на рис. 1.21а. Головная схема состоит из двух дизъюнктов, хвостовая схема – из конъюнктора и инвертора. Соответственно с выходов головной компоненты можно «снять» только наборы 00 и 11; на входной набор 00 хвостовая схема «отвечает» выходным сигналом 1; на входной набор 11 хвостовая схема «отвечает» выходным сигналом 0. Таким образом, хвостовую компоненту можно просто заменить инвертором, входом которого является выход одного из дизъюнктов (рис. 1.21б).

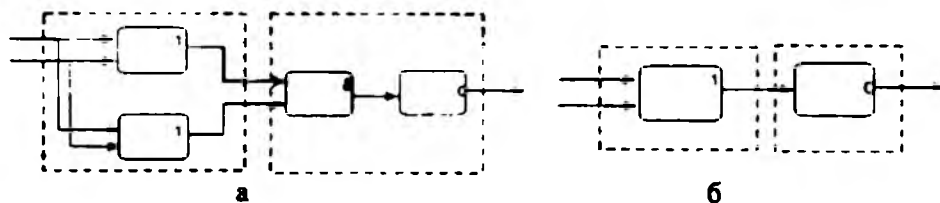


Рис. 1.21. Пример оптимизации последовательной сети из двух комбинационных схем

В общем случае может быть задана цифровая схема, содержащая несколько компонент, поведение которых описывается конечными автоматами (рис. 1.22). Для каждой компоненты требуется найти ее оптимальную в некотором смысле реализацию. Например, найти компоненту с наименьшим числом состояний или наименьшим числом связей с соседями и т.п.

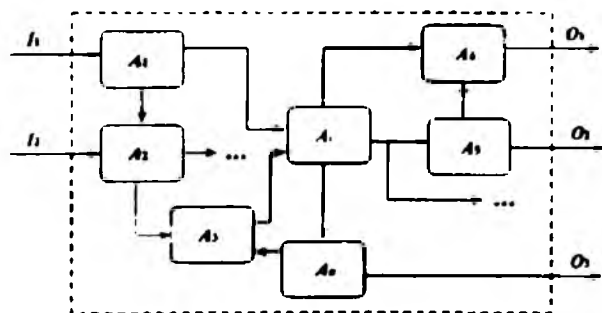


Рис. 1.22. Сеть из n взаимодействующих автоматов

Поскольку операция синхронной композиции является ассоциативной, то на практике чаще всего применяется покомпонентная итеративная оптимизация, которая может быть осуществлена на основе решения подходящего автоматного уравнения. Выбираем компоненту A_1 для оптимизации. Поведение всех остальных компонент описываем одним автоматом C , который часто называют *контекстом*, и решаем автоматное уравнение $C \bullet X = C \bullet A_1$. Из общего решения уравнения выбираем наилучшее (в интересующем нас смысле) решение A'_1 (иногда таким решением будет автомат A_1) и заменяем компоненту A_1 в автоматной сети на рис. 1.22 компонентой A'_1 , если A'_1 «лучше», чем A_1 .

После замены компоненты A_1 автоматом A'_1 , переходим к оптимизации компоненты A_2 и т.д. Процесс повторяется до тех пор, пока возможна оптимизация одной из компонент, или мы не будем удовлетворены результатами оптимизации.

В разделе 4 приведен пример такой оптимизации (рис. 4.1-4.3).

1.6.2 Логический синтез

Решение автоматных уравнений может быть использовано для синтеза устройств логического управления. Предположим, что имеется некоторое, например, механическое, устройство, объект управления (ОУ), которым нужно управлять согласно спецификации S . Примерами механических устройств могут служить станки с программным управлением, различные датчики и т.п. Требуется построить контроллер, устройство управления (УУ), который в композиции с объектом управления (ОУ) удовлетворяет заданной спецификации (рис. 1.23).

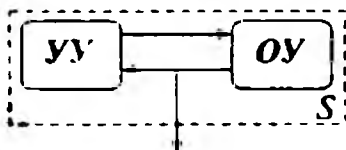


Рис 1.23 Взаимодействие устройства управления и объекта управления

Под *контроллером* (или *компенсатором*) понимается устройство, которое управляет поведением некоторого другого устройства с целью не допустить выхода последнего за определенные рамки. Данная задача может быть решена с помощью автоматного уравнения $X \bullet ОУ \equiv S$, где $ОУ$ – автомат, описывающий поведение объекта управления. S – автомат, описывающий требуемое поведение системы; часто автомат S

называют *спецификацией*. При синтезе контроллера спецификация S в общем случае может быть недетерминированным автоматом.

Если некоторая цифровая схема (дискретная система) представлена в виде сети из взаимодействующих подсистем, поведение которых описано автоматами, и синтезирована только часть такой системы (например, на рис. 1.22 отсутствует компонента A_j), то для синтеза этой компоненты можно использовать автоматное уравнение $C \bullet X \cong S$, где автомат C описывает совместное поведение компонент известной части системы, и автомат S описывает требуемое поведение всей композиции, т.е. S является автоматом-спецификацией.

Другим примером использования автоматных уравнений для логического синтеза являются конверторы для согласования протоколов в телекоммуникационных системах, поскольку разнородность существующих систем часто приводит к несоответствию протоколов. Так как для стандартизации сетевых протоколов требуется значительное количество времени, то в качестве практического решения может быть предложено установление преобразователя между несоответствующими протоколами. Если представить спецификации протоколов в виде конечных автоматов, то проблема синтеза преобразователя может быть сведена к решению соответствующего автоматного уравнения. Пример синтеза такого конвертора подробно описан в главе 4.

1.6.3. Тестирование и диагностика дискретных систем

Рассмотрим систему, поведение которой описывается автоматом на рис. 1.1а. Как всегда, в случае, когда мы хотим протестировать систему на наличие в ней неисправностей определенного класса, нам необходимо ввести модель неисправности. *Модель неисправности* обычно содержит три элемента, т.е. является тройкой $\langle S, \sim, \mathfrak{J} \rangle$. Автомат S описывает требуемое поведение тестируемой системы (эталонный автомат или автомат-спецификация), \sim – отношение, в котором должны находиться исправная проверяемая система и эталонный автомат, и \mathfrak{J} – есть множество автоматов, которые описывают поведение систем с неисправностями из заданного класса. Под *тестом* понимается конечное множество входных последовательностей конечной длины автомата-спецификации. Тест называется *полным* (относительно заданного класса неисправностей или, соответственно, относительно модели $\langle S, \sim, \mathfrak{J} \rangle$), если тест обнаруживает любую систему с неисправностью из заданного класса, т.е. любой автомат $B \in \mathfrak{J}$, который

не находится в отношении \sim с автоматом-спецификацией S , т.е. $B \neq S$. Обычно в качестве отношения соответствия \sim рассматриваются отношения эквивалентности или редукции, т.е. автомат, описывающий поведение правильно функционирующей тестируемой системы, должен делать только то, что предписано автоматом-спецификацией. Иными словами, поведение тестируемой системы на каждой входной последовательности должно совпадать с поведением (или содержаться в поведении) автомата-спецификации.

При тестировании многокомпонентных дискретных систем обычно ограничиваются тестами, которые являются полными для каждой компоненты системы как изолированного автомата. Предположим, что мы хотим синтезировать тесты для проверки функционирования компоненты B (рис. 1.20). Если эти тесты синтезированы для изолированного автомата B , то часть из них не сможет быть доставлена на компоненту B . Например, вполне возможно, что некоторые входные последовательности просто не могут быть получены на выходе компоненты A . С другой стороны, если мы не можем наблюдать выходные реакции компоненты A (нет доступа к внутренним каналам), то тесты, построенные для изолированного автомата A , могут не обнаружить интересующие нас неисправности, т.к. неправильные выходные реакции на тестовые последовательности «маскируются» компонентой B , хотя и могут проявиться на других (не тестовых) последовательностях. Более того, некоторые неисправности компоненты A могут оказаться необнаружимыми в контексте компоненты B , т.е. композиция в целом сохранит правильное поведение даже при наличии неисправности, что в свою очередь можно интерпретировать как устойчивость к неисправностям такого рода.

Для описания необнаружимых неисправностей компоненты B (A) в контексте компоненты A (B) можно использовать наибольшее решение автоматного уравнения $A \bullet X \cong A \bullet B$ (или, соответственно, уравнения $X \bullet B \cong A \bullet B$). В частности, неисправность компоненты B является необнаружимой в контексте компоненты A , если и только если в результате этой неисправности компонента B будет преобразована в автомат B' , который является решением уравнения $A \bullet X \cong A \bullet B$. Таким образом, общее решение автоматного уравнения показывает, с какой точностью возможно тестирование компоненты. Кроме того, общее решение дает возможность синтезировать компоненты автоматной сети, которые являются устойчивыми к определенным неисправностям.

В качестве других областей, в которых может быть использовано решение автоматных уравнений, можно привести криптографию (автоматные криптосистемы), логические игры (формирование выигрышной или беспроигрышной стратегии), синтез стабилизаторов для асинхронных систем. Заинтересованный читатель может найти необходимые ссылки в списке дополнительной литературы.

1.7. Контрольные вопросы к главе 1

1. Определить операцию синхронной композиции для двух автоматов.
2. Определить операцию параллельной композиции для двух автоматов.
3. Что такое синхронное автоматное уравнение?
4. Что такое синхронное автоматное неравенство? Чем оно отличается от синхронного автоматного уравнения?
5. Всегда ли разрешимо синхронное автоматное уравнение? Синхронное автоматное неравенство?
6. Может ли синхронное автоматное уравнение (неравенство) иметь два неэквивалентных решения?
7. Какой автомат называется решением синхронного автоматного уравнения?
8. Какой автомат называется наибольшим решением синхронного автоматного неравенства?
9. Каков физический смысл наибольшего решения синхронного автоматного уравнения?
10. Существует ли наименьшее решение синхронного автоматного уравнения?
11. Что такое параллельное автоматное уравнение?
12. Что такое параллельное автоматное неравенство? Чем оно отличается от параллельного автоматного уравнения?
13. Всегда ли разрешимо параллельное автоматное уравнение? Параллельное автоматное неравенство?
14. Может ли параллельное автоматное уравнение (неравенство) иметь два неэквивалентных решения?
15. Какой автомат называется решением параллельного автоматного уравнения?
16. Какой автомат называется наибольшим решением параллельного автоматного уравнения?
17. Каков физический смысл наибольшего решения параллельного автоматного уравнения?

18. Существует ли наименьшее решение параллельного автоматного уравнения?
19. Как может быть использовано решение автоматных уравнений при оптимизации логических схем?
20. Как может быть использовано решение автоматных уравнений в логическом синтезе цифровых схем?
21. Как Вы думаете, при решении каких еще задач можно использовать автоматные уравнения?

2. РЕШЕНИЕ СИНХРОННОГО АВТОМАТНОГО УРАВНЕНИЯ

2.1. Решение автоматного уравнения для последовательной композиции

В этом разделе мы обсуждаем решение синхронных автоматных уравнений и неравенств и показываем, что разрешимое синхронное автоматное уравнение всегда имеет наибольшее решение. К сожалению, не всякая даже полностью определенная редукция наибольшего решения является решением уравнения. Поэтому мы рассматриваем различные частные случаи, когда наибольшее решение полностью описывает все решения уравнения.

2.1.1. Решение автоматного уравнения для хвостовой компоненты последовательной композиции

Как мы отмечали в первой главе, впервые автоматное уравнение было явно использовано для оптимизации хвостовой компоненты последовательной автоматной сети, в которой обе компоненты суть детерминированные полностью определенные автоматы (рис. 1.20). Предлагаемый метод нахождения наибольшего решения для хвостовой компоненты был основан на вычислении выходных последовательностей головной компоненты, и было показано, что общим решением автоматного уравнения является частичный детерминированный автомат B . Поведение автомата B определено на всех последовательностях, которые являются выходными последовательностями головной компоненты A и только на них. Выходные реакции автомата B на эти последовательности совпадают с реакциями автомата B . Любой полностью определенный автомат, квазиэквивалентный автомату B , может заменить хвостовую компоненту без изменения внешнего поведения сети. Добавим в частичный автомат B новое, так называемое «безразличное состояние» DNC (don't care state), и в состоянии DNC добавим петлю, помеченную всеми входо-выходными символами автомата B . Каждый неопределенный переход в автомате B доопределим как переход в состояние DNC со всеми возможными выходными символами и обозначим *Largest* полученный полностью определенный автомат. Поскольку известно, что в этом случае множество полностью определенных редукций автомата *Largest* совпадает с множеством всех

полностью определенных автоматов, квазиэквивалентных автомату B , то автомат *Largest* является наибольшим решением автоматного уравнения $A \bullet X \cong A \bullet B$ для топологии на рис. 1.20.

Формально автомат *Largest* может быть получен следующим образом. Пусть компоненты композиции на рис. 1.20 суть автоматы $A = (A, I, U, T_A, a_0)$ и $B = (B, U, O, T_B, b_0)$. Для представления множества выходных последовательностей автомата A на первом шаге построим автомат $A_{dnc} = (A, U, O, T, a_0)$. Состояниями автомата A_{dnc} являются состояния автомата A с тем же начальным состоянием a_0 . В автомате A_{dnc} есть переход (a, u, o, a') , если и только если в автомате A есть переход (a, i, u, a') для некоторого входного символа $i \in I$. Если из пометки каждого перехода автомата A_{dnc} удалить выходные символы, то полученный полуавтомат будет представлять язык выходных последовательностей автомата A . В общем случае автомат A_{dnc} недетерминированный и ненаблюдаемый; если необходимо, то можно привести автомат A_{dnc} к наблюдаемой форме. Построим автомат B как пересечение автоматов A_{dnc} и B . В частичный автомат B добавляется новое состояние *DNC*, в котором есть петля, помеченная всеми входо-выходными символами автомата B . Каждый неопределенный переход в автомате B доопределяется как переход в состояние *DNC* со всеми выходными символами. Полученный полностью определенный автомат обозначим как *Largest*(B).

Теорема 2.1. Полностью определенные редукции автомата *Largest*(B) и только они суть решения автоматного уравнения $A \bullet X \cong A \bullet B$.

□

Иными словами, любая редукция автомата *Largest*(B), т.е. любой автомат, квазиэквивалентный автомату B , может заменить в сети $A \bullet B$ хвостовую компоненту B без изменения внешнего поведения сети.

Пример. Пусть автоматы A и B имеют диаграммы переходов, представленные на рис. 2.1. Построим автомат A_{dnc} , который будет представлять выходные последовательности автомата A (рис. 2.2а).

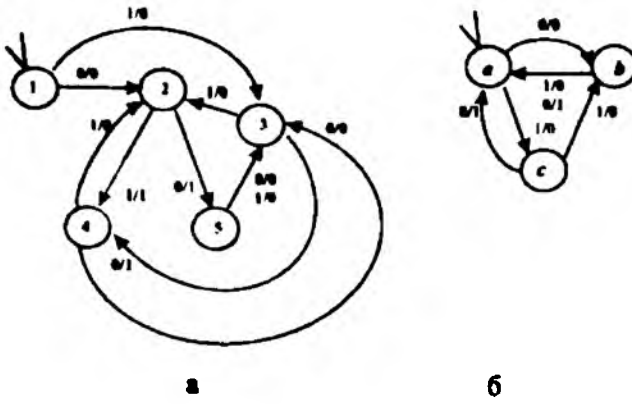


Рис. 2.1, а диаграмма переходов автомата A
 б диаграмма переходов автомата B

На следующем шаге строим пересечение B' автомата A_{chc} и автомата B (рис. 2.26).

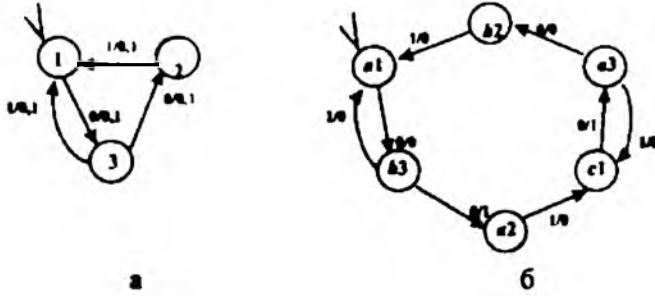


Рис. 2.2, а автомат A_{chc}
 б автомат B'

Исходную компоненту B можно заменить автоматом B' (рис. 2.3) с двумя состояниями, который квазиэквивалентен автомату B , где $a = \{a1, a2, a3\}$; $b = \{b2, b3, c1\}$.

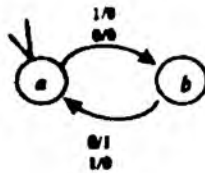


Рис. 2.3. Хвостовая компонента B' с двумя состояниями

Поскольку поведение автоматов B и B' на множестве выходных последовательностей автомата A одно и то же, то автоматы $A \bullet B'$ и $A \bullet B$ эквивалентны.

2.1.2. Решение автоматного уравнения для головной компоненты последовательной композиции

Методы решения автоматного уравнения для головной компоненты последовательной сети из полностью определенных детерминированных автоматов появились значительно позже и основывались на использовании входных последовательностей хвостовой компоненты B , которые являются эквивалентными с точки зрения внешнего поведения сети.

Две входные последовательности автомата B называются *внешне эквивалентными*, если автомат B реагирует на них одной и той же выходной последовательностью. Пусть $Largest(A)$ – наибольшее решение уравнения $X \bullet B \equiv A \bullet B$ и α – входная последовательность автомата A . Тогда множество выходных реакций автомата $Largest(A)$ на α содержит все входные последовательности автомата B , которые являются внешне эквивалентными выходной реакции автомата A на последовательность α . В этом случае любая полностью определенная редукция автомата $Largest(A)$ является решением уравнения $X \bullet B \equiv A \bullet B$. Входные последовательности автомата B , внешне эквивалентные реакции β автомата A на последовательность α , можно представить как множество реакций специального недетерминированного автомата входную последовательность β .

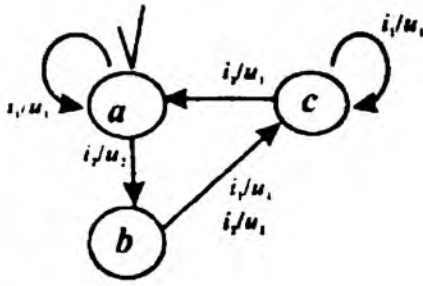
Таким образом, на первом шаге решения уравнения $X \bullet B \equiv A \bullet B$ строится недетерминированный автомат B' , множество состояний которого есть подмножество множества пар состояний автомата B с начальным состоянием (b_0, b_0) , входной и выходной алфавит совпадают с входным алфавитом автомата B . В автомате B' есть переход $((b_1, b_2), u, u', (b_1', b_2'))$, если и только если существует такой выходной символ o , что в автомате B есть переходы (b_1, u, o, b_1') и (b_2, u', o, b_2') . Первое состояние b_1 пары (b_1, b_2) «отслеживает» выходную реакцию автомата B на заданную входную последовательность βu . А второе состояние b_2 пары (b_1, b_2) добавляет к β все входные символы u' , такие, что автомат B имеет одну и ту же выходную реакцию на последовательности βu и $\beta u'$. Соответственно, построенный таким образом автомат B' в качестве реакции на любую входную последовательность β

содержит каждую входную последовательность γ такую, что реакции автомата B на β и γ совпадают. Иными словами, множество выходных последовательностей автомата B на входную последовательность β содержит все входные последовательности, внешне эквивалентные последовательности β . Для построения наибольшего решения $Largest(A)$ теперь достаточно построить последовательную композицию автоматов A и B : $Largest(A) = A \bullet B$. Множество выходных реакций автомата $Largest(A)$ на входную последовательность α есть множество всех последовательностей, которые являются внешне эквивалентными выходной последовательности $out(a_0, \alpha)$ автомата A на α .

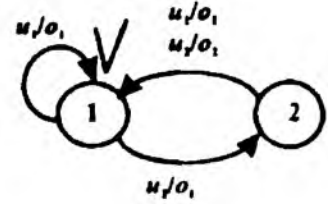
Теорема 2.2. Полностью определенные редукции автомата $Largest(A)$ и только они суть решения автоматного уравнения $X \bullet B \equiv A \bullet B$.

□

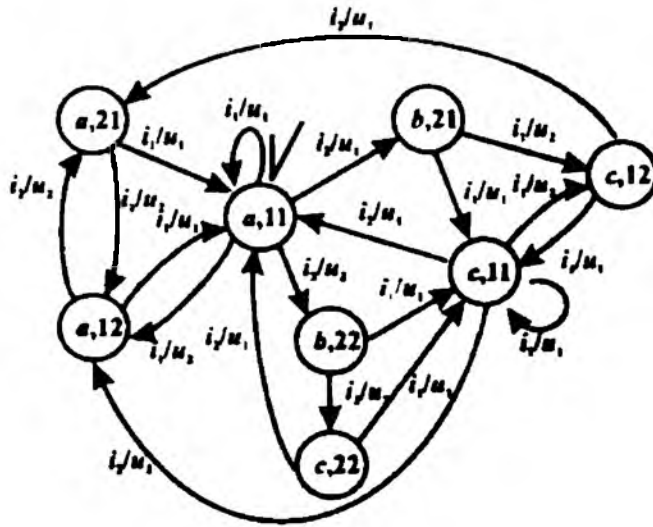
Пример. Решим автоматное уравнение $X \bullet B = A \bullet B$ для автоматов B и $A \bullet B$ на рис. 2.4а и 2.4б. Недетерминированный автомат $Largest(A)$, который будет представлять наибольшее решение уравнения $X \bullet B = A \bullet B$ приведен на рис. 2.4в. Непосредственной проверкой можно убедиться, что полностью определенная редукция автомата $Largest(A)$ с наименьшим числом состояний изоморфна автомату A , т.е. в этом примере невозможно уменьшить число состояний головной компоненты.



а



б



в

Рис. 2.4. а автомат A
 б автомат B
 в наибольшее решение $Largest(A)$
 уравнения $X \bullet B = A \bullet B$

В заключение раздела заметим, что методы решения автоматных уравнений для последовательной композиции развивались только для случая, когда автоматы-коэффициенты суть полностью определенные детерминированные автоматы. В случае, когда некоторые из коэффициентов являются недетерминированными или частичными автоматами, решение таких уравнений усложняется, однако основано обычно

на той же идее использования «безразличных» входных или выходных последовательностей. Заинтересованный читатель может найти необходимые ссылки в списке дополнительной литературы.

Задачи для самостоятельного решения

1. Для комбинационной схемы на рис. 2.5 найти множество возможных выходных векторов.

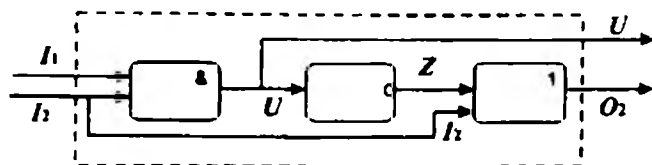


Рис. 2.5. Логическая схема

2. Для конъюнктора и дизъюнктора (рис. 1.8) найти разбиение входных векторов на множества внешне эквивалентных векторов.

3. Представить множество выходных последовательностей автомата на рис. 1.12б посредством полуавтомата.

4. Представить автомат множества внешне эквивалентных входных последовательностей для автомата на рис. 1.12в.

5. Рассмотрим последовательную композицию автоматов A и B (рис. 1.20). Диаграммы переходов автоматов A и B представлены на рис. 2.6.



Рис. 2.6, а автомат A
б автомат B

- Найти наибольшее решение автоматного уравнения $X \bullet B \cong A \bullet B$.
- Найти наибольшее решение автоматного уравнения $A \bullet X \cong A \bullet B$.

2.2. Решение автоматного уравнения для композиции с обратными связями

2.2.1. Языковой подход к решению автоматного уравнения

Наибольшее решение автоматного уравнения можно получить, выполняя операции над языками, реализуемыми этими автоматами. Рассмотрим решение автоматного уравнения $A \bullet X = S$ для автоматной сети на рис. 1.10. Известные алгоритмы нахождения наибольшего решения для композиций с обратными связями доставляют наибольшее решение соответствующего неравенства (синхронное автоматное неравенство всегда разрешимо), после чего проверяется, является ли полученное наибольшее решение решением уравнения. Поскольку операция синхронной композиции автоматов есть, вообще говоря, соответствующая композиция их языков, то для решения автоматного неравенства $A \bullet X \leq S$ достаточно найти решение языкового неравенства $((L_A^* \uparrow_{I_2 \times O_2} \cap (L_X^* \uparrow_{I_1 \times O_1}) \downarrow_{I_1 \times I_2 \times O_1 \times O_2}) \subseteq L_S^*$, в котором язык L_X^* определен в алфавите $I_2 \times O_2 \times U \times V$. Иными словами, в языке, содержащем все конечные последовательности над алфавитом $I_2 \times O_2 \times U \times V$, нужно оставить только такие последовательности, которые в композиции с языком автомата A реализуют входо-выходные последовательности автомата S . Соответственно, последовательность $\alpha \in (I_2 \times O_2 \times U \times V)^*$ может принадлежать решению неравенства $((L_A^* \uparrow_{I_2 \times O_2} \cap (L_X^* \uparrow_{I_1 \times O_1}) \downarrow_{I_1 \times I_2 \times O_1 \times O_2}) \subseteq L_S^*$, если и только если $(L_A^* \uparrow_{I_2 \times O_2} \cap \alpha \uparrow_{I_1 \times O_1}) \downarrow_{I_1 \times I_2 \times O_1 \times O_2}$ не пересекается с дополнением $\overline{L_S^*}$ языка L_S^* , т.е.

$$(L_A^* \uparrow_{I_2 \times O_2} \cap \alpha \uparrow_{I_1 \times O_1}) \downarrow_{I_1 \times I_2 \times O_1 \times O_2} \cap \overline{L_S^*} = \emptyset. \quad (2.1)$$

Выражение (2.1) эквивалентно выражению

$$\alpha \notin [(L_A^* \uparrow_{I_2 \times O_2} \cap \overline{L_S^*} \uparrow_{U \times V}) \downarrow_{I_2 \times O_2 \times U \times V}]$$

или, соответственно, выражению

$$\alpha \in \overline{\overline{(L_A^* \uparrow_{I_2 \times O_2} \cap (L_S^* \uparrow_{U \times V}) \downarrow_{I_2 \times O_2 \times U \times V})}} = \overline{L_A^* \bullet L_S^*}.$$

Таким образом, язык $L_A^* \bullet L_S^*$ есть наибольшее решение неравенства $((L_A^* \uparrow_{I_2 \times O_2} \cap (L_X^* \uparrow_{I_1 \times O_1}) \downarrow_{I_1 \times I_2 \times O_1 \times O_2}) \subseteq L_S^*$. Заметим, что в общем случае язык $L_A^* \bullet L_S^*$ не является префикс замкнутым, в то время как

язык любого конечного автомата обладает этим свойством. Поэтому наибольшим решением¹⁰ автоматного неравенства $A \bullet X \leq S$ является автомат с языком $\overline{L_A^* \bullet L_S^{pref}}$, где $\overline{L_A^* \bullet L_S^{pref}}$ есть наибольшее префикс замкнутое подмножество языка $L_A^* \bullet L_S^*$. Таким образом, для автоматного неравенства $A \bullet X \leq S$, соответствующего композиции автоматов на рис. 1.10, справедливо следующее утверждение.

Теорема 2.3. Наибольшим решением автоматного неравенства $A \bullet X \leq S$ является автомат *Largest* с языком $\overline{L_A^* \bullet L_S^{pref}}$. Более того, любая редукция автомата *Largest* также является решением автоматного неравенства $A \bullet X \leq S$. □

Еще раз отметим, что автоматное неравенство $A \bullet X \leq S$ всегда разрешимо, поскольку тривиальный автомат (автомат с пустым отношением переходов) в композиции с любым автоматом есть тривиальный автомат, т.е. является редукцией любого автомата S .

Поскольку языки, реализуемые автоматами, являются регулярными, то при построении наибольшего решения автоматного неравенства $A \bullet X \leq S$ все действия можно производить над соответствующими полуавтоматами. Ниже мы приводим алгоритм построения наибольшего решения синхронного автоматного неравенства для автоматной сети на рис. 1.10, которое затем анализируется на возможность быть таковым для автоматного уравнения.

Алгоритм 2.1. Нахождение наибольшего решения синхронного автоматного неравенства

Вход: Наблюдаемые автоматы $A = (A, I_1 \times V, U \times O_1, T_A, a_0)$ и $S = (S, I_1 \times I_2, O_1 \times O_2, T_S, s_0)$ ¹¹

Выход: Наибольшее решение *Largest* автоматного неравенства $A \bullet X \leq S$

Шаг 1. По автоматам A и S строим соответствующие полуавтоматы P_A^* и P_S^* .

¹⁰ Вообще говоря, существует не единственный автомат, который является наибольшим решением неравенства $A \bullet X \leq S$. Однако все эти автоматы эквивалентны, т.к. реализуют один и тот же язык $\overline{L_A^* \bullet L_S^{pref}}$

¹¹ Для любого автомата существует эквивалентный наблюдаемый автомат. Поэтому, если какой-то из автоматов не является наблюдаемым, то при решении уравнения можно использовать его наблюдаемую форму

Шаг 2. Строим полуавтомат $\overline{P_S^*}$: неопределенные переходы доопределяются переходом в состояние f , в состоянии f добавляется петля по всем входо-выходным парам из множества $I_1 \times I_2 \times O_1 \times O_2$, множества финальных и нефинальных состояний полуавтомата меняются местами.

Шаг 3. Строим полуавтоматы $(P_A^*)_{I_1 \times I_2 \times O_2}$ и $(P_S^*)_{U \times V}$, заменяя на каждом переходе пометку $i_1 o_1 i_2 o_2$ пометкой $i_1 i_2 o_1 o_2 u v$, $i_2 o_2 \in I_2 \times O_2$, (соответственно, пометкой $i_1 i_2 o_1 o_2 u v$, $u v \in U \times V$). Порядок алфавитов в пометках в обоих полуавтоматах одинаковый.

Шаг 4. Строим пересечение полуавтоматов $(P_A^*)_{I_1 \times I_2 \times O_2} \cap \overline{(P_S^*)_{U \times V}}$.

Шаг 5. Находим проекцию полуавтомата $(P_A^*)_{I_1 \times I_2 \times O_2} \cap \overline{(P_S^*)_{U \times V}}$ на алфавиты решения $I_2 \times O_2 \times U \times V$, заменяя на каждом переходе пометку $i_1 i_2 o_1 o_2 u v$ пометкой $i_2 o_2 u v$; детерминируем полученный полуавтомат и находим его дополнение N (неопределенные переходы доопределяем переходами в состояние DNC , в состоянии DNC добавляется петля по всем четверкам из множества $I_2 \times O_2 \times U \times V$). Удаляем все нефинальные состояния из полуавтомата N вместе с переходами из этих состояний и в эти состояния.

Шаг 6. Заменяем на каждом переходе полуавтомата N пометку $i_2 o_2 u v$ пометкой $i_2 u / o_2 v$ (т.е. разделяем символы на входные и выходные) и получаем автомат *Largest* с входным алфавитом $I_2 \times U$ и выходным алфавитом $O_2 \times V$, который и есть наибольшее решение неравенства $A \bullet X \leq S$.

□

Так как *Largest* – наибольшее решение неравенства $A \bullet X \leq S$, то наибольшее решение уравнения $A \bullet X \cong S$ совпадает с *Largest*, если и только если композиция $A \bullet \textit>Largest$ эквивалентна автомату S .

Теорема 2.4. Пусть *Largest* – наибольшее решение неравенства $A \bullet X \leq S$. Если композиция $A \bullet \textit>Largest$ эквивалентна автомату S , то *Largest* – наибольшее решение уравнения $A \bullet X \cong S$, в противном случае автоматное уравнение $A \bullet X \cong S$ неразрешимо.

Пример. Построим наибольшее решение автоматного уравнения $A \bullet X \cong S$ для автоматной сети на рис. 2.7а. Диаграммы переходов автоматов A и S представлены на рис. 2.7б и 2.7в.

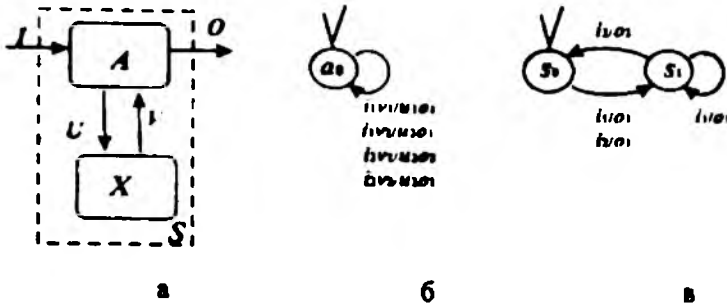


Рис. 2.7, а автоматная сеть
 б автомат А
 в автомат S

По автоматам А и S строим соответствующие полуавтоматы P_A^* и P_S^* (рис. 2.8а, б). Строим дополнение полуавтомата $\overline{P_S^*}$, где неопределенные переходы доопределяем переходом в состояние f , финальные и нефинальные состояния меняются местами (нефинальные состояния заштрихованы). Строим полуавтомат $(P_S^*)_{\uparrow U, V}$ (рис. 2.8в), расширяя полуавтомат $\overline{P_S^*}$ на алфавиты U и V ¹².

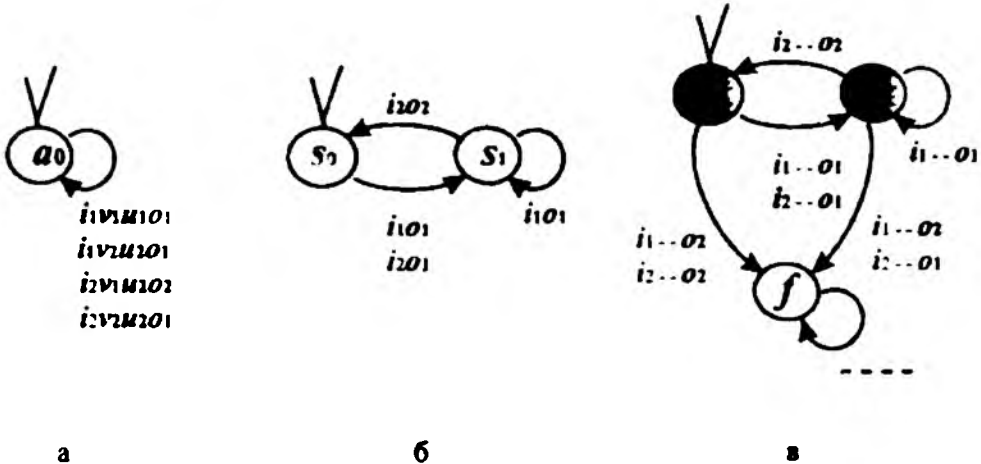


Рис. 2.8, а полуавтомат P_A^*
 б полуавтомат P_S^*
 в полуавтомат $(P_S^*)_{\uparrow U, V}$.

¹² Символ «-» используется для представления всех символов из соответствующего алфавита

Пересечение полуавтоматов P_A^* и $(\overline{P_S^*})_{U \times V}$ показано на рис. 2.9а, проекция пересечения на алфавиты решения $U \times V$ представлена на рис. 2.9б ($b_0 = a_0s_0, b_1 = a_0s_1, f = a_0f$).

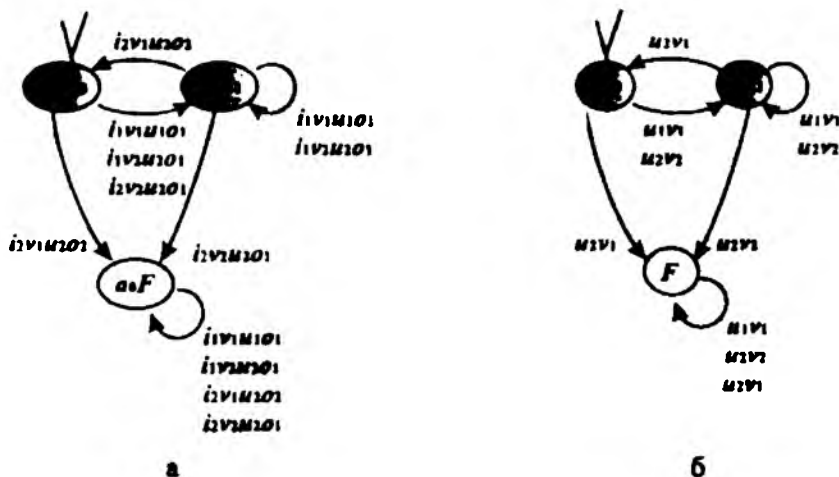


Рис. 2.9, а полуавтомат $P_A^* \cap (\overline{P_S^*})_{U \times V}$

$$б (P_A^* \cap (\overline{P_S^*})_{U \times V})_{U \times V}$$

Полуавтомат $(P_A^* \cap (\overline{P_S^*})_{U \times V})_{U \times V}$ является недетерминированным, поэтому на следующем шаге детерминизируем его и строим дополнение (рис. 2.10а). Неопределенные переходы доопределяем переходами в состояние *DNC*, финальные и нефинальные состояния меняются местами. Из построенного полуавтомата удаляем все нефинальные состояния и переходы в них. Затем по полуавтомату строим соответствующий ему автомат *Largest* с входным алфавитом U и выходным алфавитом V (рис. 2.10б).

Композиция автоматов *Largest* и A эквивалентна спецификации S , следовательно, *Largest* – наибольшее решение автоматного уравнения.

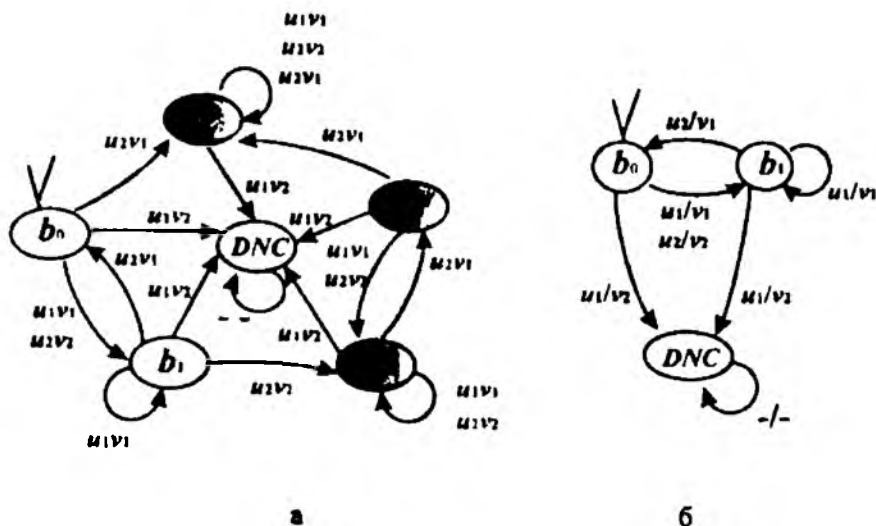


Рис. 2.10, а полуавтомат $(P_A \cap (P_S^*)_{(U,V)})_{(U,V)}$
 б автомат *Largest*

Заметим, что разрешимость автоматного уравнения существенно зависит от алфавитов неизвестной компоненты и существуют в некотором смысле наибольшие входной и выходной алфавиты, для которых имеет смысл искать решение. При решении уравнения для композиции на рис. 2.11 такими входным и выходным алфавитами являются $I_1 \times I_2 \times O_1 \times U$ и $O_2 \times V$. Если не существует решения уравнения с такими алфавитами, то не существует решения с алфавитами $I_2 \times U$ и $O_2 \times V$. Обратное верно не всегда. Если не существует решения уравнения с алфавитами $I_2 \times U$ и $O_2 \times V$, то вполне возможно, что существует решение с входным алфавитом $I_1 \times I_2 \times O_1 \times U$ и выходным алфавитом $O_2 \times V$. В следующем разделе мы описываем алгоритм построения наибольшего решения автоматного неравенства $A \bullet X \leq S$ (уравнения $A \bullet X \cong S$), в котором отсутствуют операции дополнения и детерминизации полуавтоматов. Однако данный алгоритм применим только в том случае, когда автомат A является полностью определенным детерминированным автоматом.

2.2.2. Алгоритм построения наибольшего решения в наибольшем алфавите

В случае, когда автомат A является полностью определенным детерминированным автоматом, нас интересует решение уравнения для

композиции на рис. 1.10, и мы не очень озабочены минимальностью алфавитов неизвестной компоненты, наибольшее решение автоматного неравенства $A \bullet X \leq S$ (уравнения $A \bullet X \cong S$) можно получить без использования операций дополнения и детерминизации полуавтоматов. В этом случае мы строим решение в алфавитах $I_1 \times I_2 \times O_1 \times U$ и $O_2 \times V$ (т.е. во всех алфавитах, которые имеются в автоматной сети), и, соответственно, построенная автоматная сеть (рис. 2.11) будет иметь несколько иную структуру, чем на рис. 1.10.

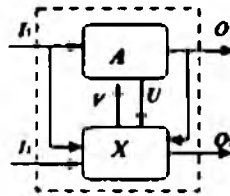


Рис. 2.11. Композиция автоматов A и B

Алгоритм 2.2. Нахождение наибольшего решения автоматного неравенства в наибольшем алфавите

Вход: Полностью определенные детерминированные автоматы $A = (A, I_1 \times V, U \times O_1, T_A, a_0)$ и $S = (S, I_1 \times I_2, O_1 \times O_2, T_S, s_0)$

Выход: Наибольшее решение $Largest_{\max}$ автоматного неравенства $A \bullet X \leq S$ с входным алфавитом $I_1 \times I_2 \times O_1 \times U$ и выходным алфавитом $O_2 \times V$

Шаг 1. Строим полуавтоматы $(P_A^*)_{\uparrow I_2 \times O_2}$ и $(P_S^*)_{\uparrow V \times U}$, заменяя на каждом переходе пометку $i_1 o_1 u v$ ($i_1 i_2 o_1 o_2$) пометкой $i_1 i_2 o_1 o_2 u v$, $i_2 o_2 \in I_2 \times O_2$, (соответственно, пометкой $i_1 i_2 o_1 o_2 u v$, $u v \in U \times V$). Порядок алфавитов в пометках в обоих полуавтоматах одинаковый. Строим пересечение полуавтоматов $(P_A^*)_{\uparrow I_2 \times O_2}$ и $(P_S^*)_{\uparrow V \times U}$.

Шаг 2. Автомат $Largest_{\max}$ имеет то же множество состояний, что и пересечение $(P_A^*)_{\uparrow I_2 \times O_2} \cap (P_S^*)_{\uparrow V \times U}$ в объединении со специальным безразличным состоянием DNC . В автомате $Largest_{\max}$ существует переход $(as, i_1 i_2 o_1 u, o_2 v, d's')$, если и только если в полуавтомате $(P_A^*)_{\uparrow I_2 \times O_2} \cap (P_S^*)_{\uparrow V \times U}$ есть переход $(as, i_1 i_2 o_1 o_2 u v, d's')$. В автомате $Largest_{\max}$ существует переход $(as, i_1 i_2 o_1 u, o_2 v, DNC)$, если и только если в состоянии a автомата A отсутствует переход с входо-выходной парой $i_1 v / o_1 u$. В состоянии DNC присутствует петля, помеченная всеми входо-выходными символами.

Теорема 2.5. Пусть $Largest_{max}$ – наибольшее решение неравенства $A \bullet X \leq S$ для композиции на рис. 2.11. Если композиция $A \bullet Largest_{max}$ эквивалентна автомату S , то $Largest_{max}$ – наибольшее решение уравнения $A \bullet X \equiv S$ для автоматной сети на рис. 2.11; в противном случае автоматное уравнение $A \bullet X \equiv S$ неразрешимо как для композиции на рис. 2.11, так и для композиции на рис. 1.10. □

Пример. Рассмотрим композицию автоматов на рис. 2.12а.

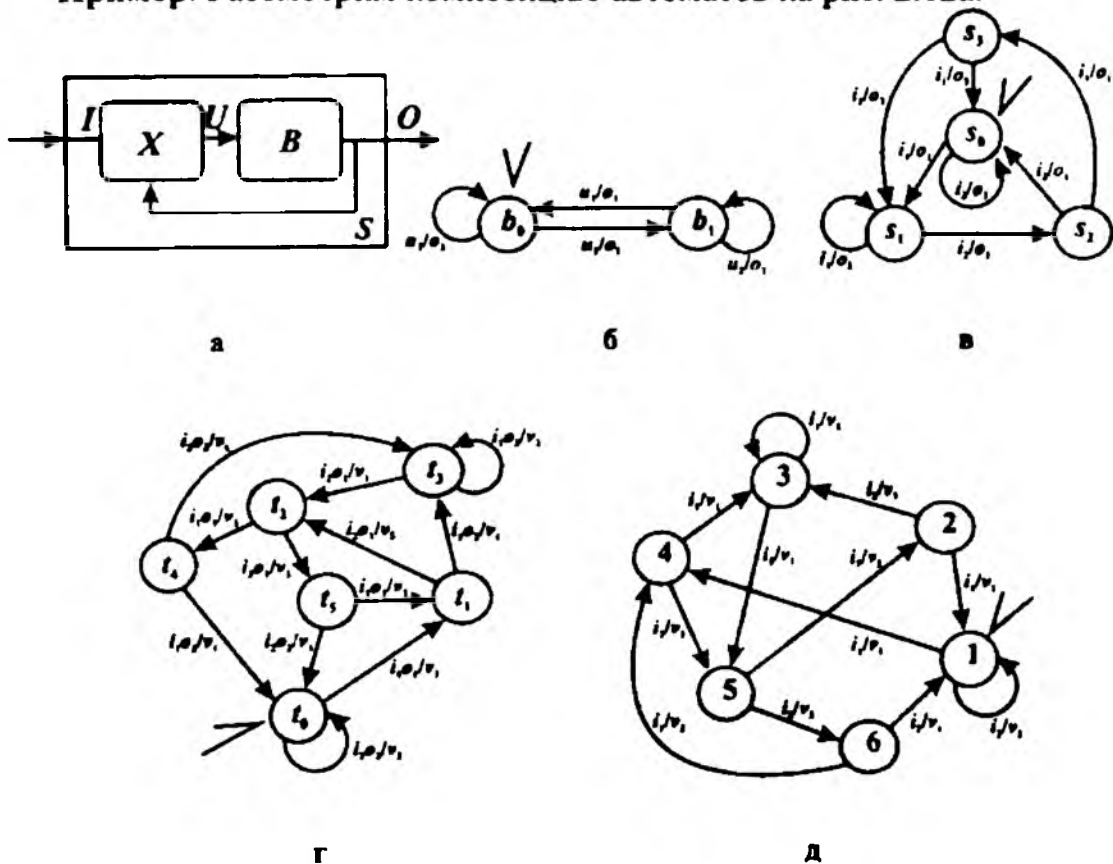


Рис. 2.12, а автоматная сеть
 б диаграмма переходов автомата B
 в диаграмма переходов автомата S
 г одно из решений уравнения $X \bullet B \equiv S$
 д решение уравнения для композиции на рис. 1.20

Диаграммы переходов автоматов B и S приведены на рис. 2.12б и 2.12в соответственно. Одно из решений уравнения $X \bullet B \equiv S$ приведено

на рис. 2.12г¹³. В данном случае существует решение уравнения $X \bullet B \cong S$ и для последовательной композиции на рис. 1.20, и это решение показано на рис. 2.12д.

Сравнивая алгоритмы 2.1 и 2.2, можно сделать следующие замечания.

1) В отличие от алгоритма 2.1, алгоритм 2.2 имеет полиномиальную сложность относительно числа состояний автоматов A и S . Экспоненциальная сложность алгоритма 2.1 связана с операцией детерминизации проекции пересечения.

2) С другой стороны, решение, которое находит алгоритм 2.1, можно получить, если «спроецировать» полученный в результате алгоритма 2.2 автомат на входной алфавит $I_2 \times U$. Однако в общем случае операция проекции имеет экспоненциальную сложность относительно числа состояний проецируемого автомата.

В заключение раздела мы отмечаем две причины, по которым автоматные уравнения явно не используются при решении ряда технических задач. Во-первых, сложность решения произвольного автоматного уравнения (алгоритм 2.1) является экспоненциальной, т.к. на шаге 5 используется детерминизация полуавтомата. Во-вторых, известно, что в общем случае не всякая даже полностью определенная редукция наибольшего решения является решением автоматного уравнения $A \bullet X \cong S$; более того, до сих пор не известно эффективного метода характеристики всех редукций наибольшего решения, являющихся решениями автоматного уравнения. В следующем разделе мы предлагаем находить только часть (редукцию) наибольшего решения на основе использования безразличных последовательностей, т.е. в общем случае такое решение не является наибольшим решением уравнения. Однако, во-первых, сложность предлагаемого алгоритма является полиномиальной (относительно числа состояний автоматов-коэффициентов уравнения), а, во-вторых, любая полностью определенная редукция построенного автомата будет решением уравнения.

2.2.3. Решение автоматного уравнения на основе безразличных последовательностей

В этом разделе мы показываем, как построить часть наибольшего решения автоматного уравнения на основе безразличных последова-

¹³ Отсутствующие на рис 2.12г переходы суть переходы в безразличное состояние DNC .

тельность, т.е. последовательностей, которые не могут поступить на компоненту B в данной сети (рис. 1.10). Если автомат A детерминированный, то полученное решение будет частичным детерминированным автоматом. Любой автомат, квазиэквивалентный построенному частичному автомату, является решением автоматного уравнения. Заметим, что алгоритм основан на расширении известного решения уравнения, т.е. предполагается, что уравнение разрешимо и, более того, известно некоторое решение уравнения. Мы иллюстрируем предлагаемый подход для автоматной сети на рис. 2.7а, где $A = (A, I \times V, U \times O, T_A, a_0)$ и $B = (B, U, V, T_B, b_0)$.

Покажем, как можно построить множество безразличных последовательностей для автомата B . Множество возможных входных последовательностей автомата B есть множество последовательностей в алфавите U , вырабатываемых автоматом A . Таким образом, безразличными входными последовательностями для компоненты B будут последовательности в алфавите U , которые никогда не могут появиться на выходе автомата A в соответствующем состоянии. Поведение автомата B на таких последовательностях может быть любым.

Алгоритм 2.3. Нахождение безразличных входных последовательностей для компоненты

Вход: Полностью определенные детерминированные автоматы $A = (A, I \times V, U \times O, T_A, a_0)$ и $B = (B, U, V, T_B, b_0)$ и композиция автоматов $N = A \bullet B$ (рис. 2.7а).

Выход: Частичный автомат B'

Шаг 1. Для каждого состояния $a_j \in A$ такого, что существует $b \in B$ со свойством $(a_j, b) \in N$, находим множество D_{a_j} безразличных символов в алфавите U , то есть множество символов, которые не могут появиться на выходе автомата A в состоянии a_j . Символ u_m принадлежит D_{a_j} , если и только если для любых $i \in I, v \in V, o \in O$ и $a_k \in A$ справедливо $(a_j, iv, u_m o, a_k) \notin T_A$. Множество D_b безразличных входных символов $u_m \in U$ в состоянии $b \in B$ есть пересечение множеств D_{a_j} по всем $(a_j, b) \in N, a_j \in A$.

Шаг 2. Для каждого состояния $b \in B$ в автомате оставляем переходы только для входных символов $u \in U \setminus D_b$. В результате получаем частичный автомат B' .

Теорема 2.6. Любой автомат, квазиэквивалентный автомату B , является решением автоматного уравнения $A \bullet X \equiv A \bullet B$, т.е. может заменить компоненту B без изменения поведения композиции.

□

Пример. Рассмотрим автоматную сеть на рис. 2.7а. Диаграммы переходов автоматов A и B приведены на рисунке 2.13.

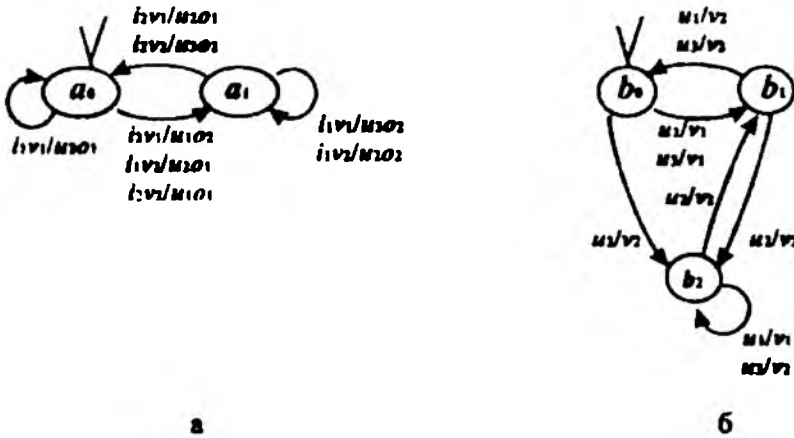


Рис. 2.13, а автомат A
б автомат B

Строим автомат $N = A \bullet B$ (рис. 2.14) с входным алфавитом I и выходным алфавитом O . Множество состояний автомата N есть $\{a_0b_0, a_1b_1, a_0b_2, a_1b_2\}$.

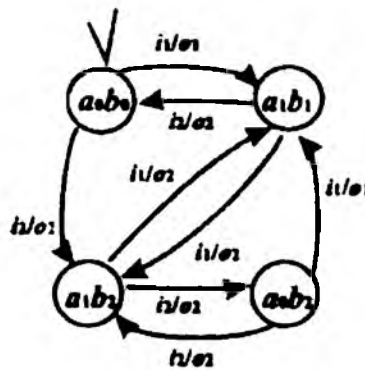


Рис 2.14. Автомат N

Для каждого состояния автомата A находим множество безразличных символов в алфавите U . Для состояния a_0 множество $D_{a_0} = \{u_3\}$, так как в состоянии a_0 никогда не появляется внутренний выходной символ u_3 , аналогично $D_{a_1} = \{u_1\}$. Проверяем все состояния автомата B . В автомате N состояние b_0 встречается только совместно с состоянием

a_0 . Так как $D_{a_3} = \{u_3\}$, то в автомате B переход из состояния b_0 по символу u_3 заменяем неопределенным переходом. Состояние b_1 встречается только совместно с состоянием a_1 , поэтому переход по безразличному символу u_1 заменяем неопределенным переходом. Состояние b_2 встречается совместно с состояниями a_0 и a_1 . Находим пересечение множеств безразличных символов $D_{a_0} \cap D_{a_1} = \{u_3\} \cap \{u_1\} = \emptyset$, поэтому в состоянии b_2 не появляется неопределенных переходов. Построенный частичный автомат B представлен на рисунке 2.15а.

Любой автомат, квазиэквивалентный автомату B , может заменить компоненту B без изменения внешнего поведения сети. В качестве оптимальной компоненты, к примеру, может быть выбран автомат B_1 с одним состоянием (рис. 2.15б), квазиэквивалентный автомату B .

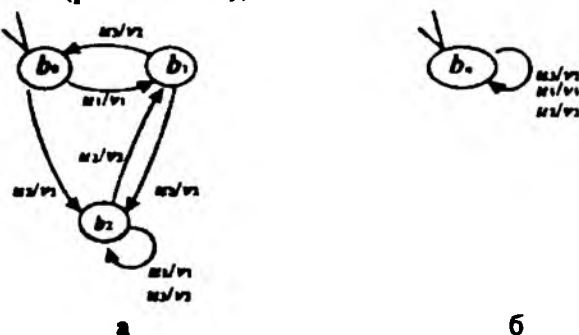


Рис. 2.15, а автомат B
б автомат B_1

Заметим, что в общем случае нельзя упростить операцию построения частичного автомата B , взяв проекцию пересечения $L_A \cap (L_B)_{\{1,1,0\}}$ на алфавит $U \times V$. Можно показать, что построенный таким образом автомат B в качестве квази-эквивалентных автоматов может содержать автоматы, которые не являются решениями уравнения.

Задачи для самостоятельного решения

1. Найти наибольшее решение уравнения $X \bullet B \cong S$ для композиции на рис. 2.12а и автоматов B и S на рис. 2.16.
2. Найти наибольшее решение уравнения $X \bullet B \cong S$ для композиции (рис. 1.20) для автоматов, диаграммы переходов которых представлены на рис. 2.16. Сравнить с решением, полученным для упражнения 1.

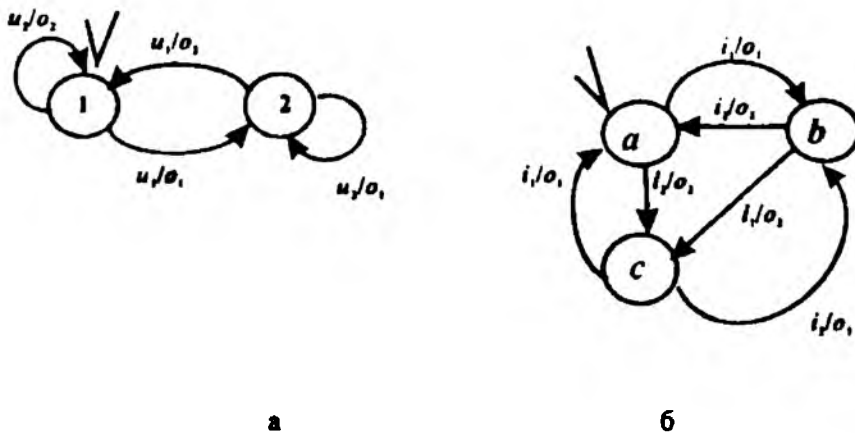


Рис. 2.16, а диаграмма переходов автомата В
б диаграмма переходов автомата S

3. Найти решения для упражнений 1 и 2 с использованием безразличных последовательностей и сравнить с решениями, полученными ранее.

2.3. Частные решения синхронного автоматного уравнения

Как уже отмечалось, с практической точки зрения наиболее интересны полностью определенные решения автоматного уравнения. Причиной является тот факт, что поведение практически всех дискретных систем описывается полностью определенными автоматами. В частности, при логическом синтезе цифровых схем, только полностью определенные решения можно использовать для синтеза неизвестной компоненты, поскольку поведение логической схемы определено на любой входной последовательности. Если соответствующее уравнение не имеет полностью определенного решения, то это означает, что ни одну логическую схему нельзя добавить к уже синтезированной части для того, чтобы получить требуемое внешнее поведение. Кроме того, после нахождения наибольшего решения хотелось бы достаточно просто охарактеризовать редукции наибольшего решения, которые также являются решениями уравнения. В данном разделе мы рассматриваем частные решения автоматного уравнения, такие как полностью определенные решения и муровские решения. В последнем разделе мы кратко обсуждаем, какие еще частные решения могут представлять теоретический или практический интерес, и какие из таких частных решений достаточно хорошо исследованы.

2.3.1. Полностью определенные решения синхронного автоматного уравнения

Наибольшее решение *Largest* разрешимого автоматного уравнения $A \bullet X \equiv S$ является в общем случае частичным недетерминированным автоматом. Наибольшее *полностью определенное* решение *L* автоматного уравнения (если существует) есть полностью определенный автомат, который в качестве редукиций содержит все полностью определенные решения автоматного уравнения. По определению, *L* есть редукиция автомата *Largest*. Обозначим через $Com_{Largest}$ наибольший полностью определенный подавтомат автомата *Largest*. Согласно теореме 3.3 в части 1 настоящего пособия, любая полностью определенная редукиция автомата *Largest* есть также редукиция его наибольшего полностью определенного подавтомата $Com_{Largest}$. Таким образом, автомат *L* (если существует) совпадает с автоматом $Com_{Largest}$.

Если *Largest* – полностью определенный автомат, то *Largest* – наибольшее полностью определенное решение автоматного уравнения. Иначе последовательно удаляем в автомате *Largest* состояния, в которых не определен хотя бы один переход, и переходы в эти состояния. Если удаляется начальное состояние, то уравнение не имеет полностью определенного решения. Иначе полученный полностью определенный автомат $Com_{Largest}$ есть наибольшее полностью определенное решение *L* автоматного уравнения.

Теорема 2.7. Пусть *Largest* – наибольшее решение уравнения. Наибольшее полностью определенное решение автоматного уравнения $A \bullet X \equiv S$ совпадает с наибольшим полностью определенным подавтоматом $Com_{Largest}$ автомата *Largest*. Если автомат *Largest* не имеет полностью определенных подавтоматов, то уравнение не имеет полностью определенного решения.

□

Вообще говоря, если автоматное уравнение не имеет полностью определенного решения, то такое уравнение должно использоваться очень аккуратно, т.к., как отмечалось выше, физические реализации обычно являются полностью определенными. Частичность наибольшего решения показывает, что при любом доопределении неопределенного перехода в наибольшем решении композиция доопределенного решения с автоматом *A* будет иметь входо-выходные последовательности, которых нет в спецификации *S*. Проблема может быть ре-

шена путем удаления соответствующих входных последовательностей из автомата-спецификации S ; однако при этом автомат S получится частичным, и проектировщик должен быть уверен, что рабочими последовательностями композиции являются только последовательности, на которых определено поведение автомата S .

Пример. Рассмотрим композицию автоматов на рис. 2.20 и решим автоматное уравнение для $X \bullet B \cong S$ для автоматов B и S , диаграммы которых приведены на рис. 2.17а и 2.17б. Наибольшее решение показано на рис. 2.17в. Непосредственной проверкой можно убедиться, что в состоянии $b1$ не определено ни одного перехода. Соответственно, мы удаляем это состояние и все переходы в это состояние. После этого в состоянии $a3$ становится неопределенным переход под действием входного символа i_2 . Удаляем состояние $a3$ и получаем наибольшее полностью определенное решение (рис. 2.17г).

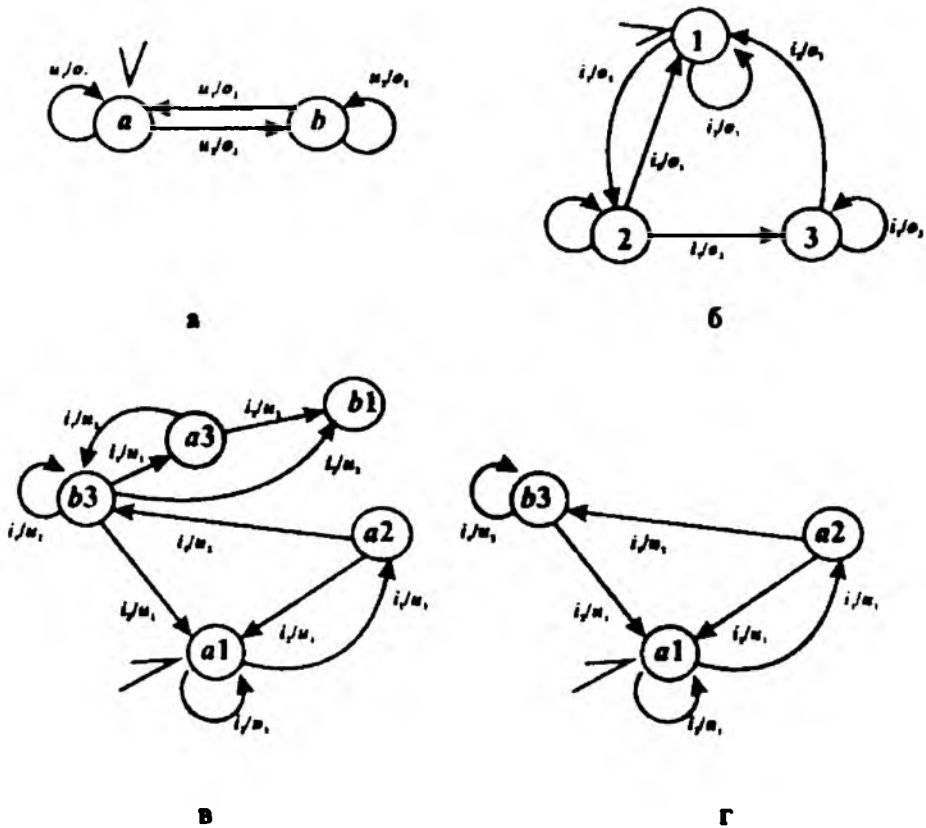


Рис. 2.17. а автомат B
 б автомат S
 в наибольшее решение уравнения $X \bullet B \cong S$
 г наибольшее полностью определенное решение уравнения $X \bullet B \cong S$

2.3.2. Муровские решения синхронного автоматного уравнения

Даже если автоматное уравнение имеет наибольшее полностью определенное решение, то известно, что не всякая полностью определенная редукция такого решения также является решением уравнения. Исключением являются уравнения, например, для последовательной автоматной сети (рис. 1.20) с полностью определенными детерминированными автоматами-коэффициентами.

Теорема 2.8. Для последовательной автоматной сети, в которой автоматы A и S (автоматы B и S) являются полностью определенными детерминированными автоматами, любая полностью определенная редукция наибольшего решения автоматного уравнения $A \bullet X \equiv S$ ($X \bullet B \equiv S$) для любой компоненты сети является решением уравнения. \square

Для композиций с обратными связями, где каждый автомат является полностью определенным и детерминированным, можно выделить частные случаи, когда полностью определенная редукция наибольшего решения является решением уравнения. Эта ситуация хорошо известна проектировщикам цифровых схем. Для того чтобы исключить ненужную осцилляцию, каждый цикл в синхронных цифровых схемах содержит автомат Мура. Можно сформулировать следующее утверждение для автоматного уравнения, соответствующего композиции на рис. 1.10.

Теорема 2.9. 1. Если A и S – полностью определенные детерминированные автоматы, и A есть автомат Мура, то любая полностью определенная редукция наибольшего решения уравнения $A \bullet X \equiv S$ является решением уравнения. 2. Если A и S – полностью определенные детерминированные автоматы, то любая полностью определенная детерминированная редукция наибольшего решения уравнения $A \bullet X \equiv S$, которая является автоматом Мура, является решением уравнения. \square

Подобно тому, как было построено наибольшее полностью определенное решение L автоматного уравнения $A \bullet X \equiv S$, можно построить наибольшее полностью определенное муровское решение уравнения, т.е. подавтомат L_{moore} автомата L такой, что автомат Мура является решением уравнения, если и только если этот автомат есть редукция автомата L_{moore} .

Алгоритм 2.4. Построение наибольшего полностью определенного муровского решения уравнения $A \bullet X \equiv S$

Вход: Полностью определенные детерминированные автоматы A и S и наибольшее полностью определенное решение L уравнения $A \bullet X \equiv S^{14}$

Выход: Наибольшее муровское решение L_{moore} (если существует)

Шаг 0. Полагаем $L_{moore} := L$

Шаг 1. Если в каждом состоянии автомата L_{moore} множества выходных символов для любых двух входных символов совпадают, то L_{moore} – наибольшее полностью определенное муровское решение автоматного уравнения. **КОНЕЦ.** Иначе для каждого состояния l автомата L_{moore} выполняем следующее. Определяем множество $K_1 \subseteq O_2 \times V$. Выходной символ $(o, v) \in K_1$, если для каждого входного символа $(i, u) \in I_2 \times U$ существует состояние l' автомата L_{moore} такое, что четверка $(l, (i, u), (o, v), l')$ является переходом в автомате L_{moore} . Если K_1 – пустое множество, то полагаем все переходы в состоянии l неопределенными.

Шаг 2. Последовательно удаляем в автомате L_{moore} состояния, в которых не определен хотя бы один переход, и переходы в эти состояния. Если удаляется начальное состояние, то среди решений уравнения нет ни одного автомата Мура. **КОНЕЦ.** Иначе **Шаг 1.**

□

Теорема 2.10. Автомат Мура B является решением автоматного уравнения $A \bullet X \equiv S$, если и только если B есть редукция наибольшего полностью определенного муровского решения L_{moore} .

□

Следствие. Если автоматное уравнение $A \bullet X \equiv S$ не имеет наибольшего полностью определенного муровского решения, то ни одно решение уравнения не является автоматом Мура.

□

Существует ряд других частных решений, которые интересны с теоретической и/или практической точек зрения. Некоторые из них мы кратко рассматриваем в следующем разделе.

¹⁴ Согласно определению (глава 1), автомат Мура есть полностью определенный автомат. Поэтому, если уравнение не имеет полностью определенных решений, то уравнение не имеет решений, которые являются автоматами Мура

2.3.3. Другие частные решения синхронного автоматного уравнения

Существуют и другие частные решения автоматного уравнения, которые представляют интерес с точки зрения теории и/или практики. В первую очередь интересны решения, которые в композиции с автоматом A обеспечивают полностью определенную композицию. С другой стороны, с практической точки зрения интересны решения, которые не допускают тупиков и/или осцилляций в композиции с автоматом A . Ниже мы очень коротко представляем такие частные решения; для более подробного изучения частных решений автоматного уравнения заинтересованный читатель может обратиться к дополнительной литературе.

Решение B уравнения $A \bullet X \equiv S$ называется *максимально определенным*, если композиция $A \bullet B$ определена на любой входной последовательности.

Решение B уравнения $A \bullet X \equiv S$ называется *прогрессивным*, если для любой последовательности γ из языка $(L_A) \uparrow_{I_2 \times O_2} \cap (L_B) \uparrow_{I_1 \times O_1}$ и любого $i_1 i_2 \in I_1 \times I_2$ в языке $(L_A) \uparrow_{I_2 \times O_2} \cap (L_B) \uparrow_{I_1 \times O_1}$ существует продолжение последовательности γ с $(I_1 \times I_2)$ -проекцией $i_1 i_2$.

Иными словами, решение B уравнения $A \bullet X \equiv S$ не является прогрессивным, если существует входная последовательность $\alpha(i_1 i_2)$, такая, что под действием входной последовательности α композиция $A \bullet B$ может перейти в состояние, в котором компоненты не имеют соответствующих внутренних сигналов для входного символа $i_1 i_2$. Прогрессивное решение гарантирует, что при подаче входной последовательности α , в композиции всегда существует пара внутренних сигналов, на которых компоненты могут «согласиться», т.е. после подачи входного символа «стабилизация» композиции не будет бесконечной.

По определению, любое прогрессивное решение является максимально определенным, но не наоборот. В общем случае максимально определенное решение не обязательно является прогрессивным. Известно, что если уравнение имеет прогрессивное решение, то уравнение имеет и наибольшее прогрессивное решение, т.е. прогрессивное решение, которое содержит каждое прогрессивное решение как редукцию. Однако в отличие от наибольшего полностью определенного решения уравнения и наибольшего полностью определенного муров-

ского решения уравнения, которые являются подавтоматами наибольшего решения уравнения (если существуют), наибольшее прогрессивное решение не является подавтоматом наибольшего решения уравнения. Алгоритм построения наибольшего прогрессивного решения намного сложнее, и заинтересованный читатель может найти этот алгоритм в дополнительной литературе. Заметим также, что в общем случае не любая редукция наибольшего прогрессивного решения является (прогрессивным) решением уравнения. Однако известны необходимые и достаточные условия, при которых редукция наибольшего прогрессивного решения обладает свойством прогрессивности.

Особый интерес представляют решения, когда для каждого входного символа существует единственная пара согласованных внутренних символов. Мы называем такие решения безопасными.

Решение B уравнения $A \bullet X \equiv S$, где A – детерминированный автомат, называется безопасным, если для любой входной последовательности язык $(L_A) \uparrow_{12 \times 02} \cap (L_B) \uparrow_{11 \times 01}$ содержит единственную последовательность (α, γ) для подходящей последовательности $\gamma \in (O_1 \times O_2 \times V \times U)^*$. Заметим, что каждое муровское решение является безопасным. Каждое безопасное решение является прогрессивным, но не наоборот.

Задачи для самостоятельного решения

1. Найти наибольшее полностью определенное решение для автоматного уравнения $X \bullet B \equiv S$ для композиции на рис. 2.12а с автоматами B и S на рис. 2.18.

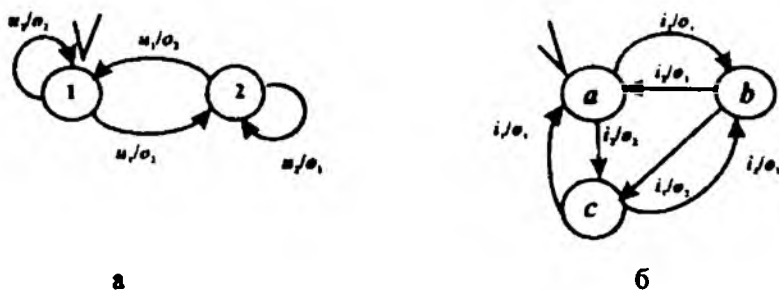


Рис. 2.18, а автомат B
б автомат S

2. Существует ли полностью определенное решение автоматного уравнения $X \bullet B \equiv S$ для топологии на рис. 1.20 с автоматами B и S на

рис. 2.18? Можно ли найти полностью определенный автомат, который в композиции с автоматом B эквивалентен автомату S ?

3. Какие входные последовательности нужно удалить из спецификации в предыдущем упражнении для того, чтобы уравнение имело полностью определенное решение? Можно ли подать на полученную композицию входную последовательность $i_1i_2i_1$? Входную последовательность $i_2i_2i_1$?

2.4. Разрешимость синхронного автоматного уравнения относительно различных топологий

Мы уже отмечали, что существуют две большие проблемы, которые не позволяют активно использовать автоматные уравнения в различных приложениях. Во-первых, сложность решения автоматного уравнения в общем случае является экспоненциальной относительно числа состояний автоматов-коэффициентов уравнения, а во-вторых, не всякая полностью определенная редукция наибольшего решения является решением уравнения. Как показано выше, такие проблемы не возникают при решении уравнений для композиций без обратных связей. В этом разделе мы рассматриваем специальную композицию с обратными связями, для которой решение уравнения $X \bullet B \equiv S$ также не имеет этих недостатков, если коэффициенты уравнения являются детерминированными полностью определенными автоматами. Соответственно, при решении задач, в которых можно самостоятельно выбирать топологию для решения уравнения (например, при локальной оптимизации цифровых схем), автоматные уравнения можно использовать достаточно эффективно.

Рассмотрим композицию на рис. 2.19, для которой нужно решить автоматное уравнение относительно компоненты X . Эта композиция обладает следующим свойством: все, в том числе внутренние, входы неизвестной компоненты являются наблюдаемыми. Кроме того, мы полагаем, что B и S являются детерминированными полностью определенными автоматами.

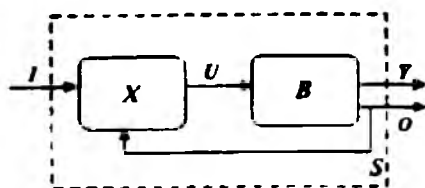


Рис. 2.19. Автоматная композиция

В разделе 2.2.2 мы показали, что если уравнение $X \bullet B \equiv S$ разрешимо, то это уравнение разрешимо для неизвестной компоненты с входным алфавитом $I \times O \times Y$ и выходным алфавитом U . Мы модифицируем алгоритм 2.2: в случае полностью определенных детерминированных автоматов B и S представленный ниже алгоритм доставляет наибольшее решение L уравнения для автоматной сети на рис. 2.19, причем множество всех полностью определенных детерминированных редуций автомата L совпадает с множеством всех полностью определенных детерминированных решений уравнения $X \bullet B \equiv S$.

Алгоритм 2.5. Нахождение решения автоматного уравнения $X \bullet B \equiv S$, полностью определенные детерминированные редуции которого и только они являются решениями уравнения

Вход: Полностью определенные детерминированные автоматы $B = (B, U, O \times Y, T_B, b_0)$ и $S = (S, I, O \times Y, T_S, s_0)$, автоматная сеть на рис. 2.19

Выход: Решение L автоматного уравнения $X \bullet B \equiv S$ с входным алфавитом $I \times O$ и выходным алфавитом U , полностью определенные детерминированные редуции которого и только они являются решениями уравнения

Шаг 1. Строим полуавтоматы $(P_B^*)_{\uparrow I}$ и $(P_S^*)_{\uparrow U}$, заменяя на каждом переходе пометку $iuoy$ (ioy) пометкой $iuoy$, $i \in I$, (соответственно, $u \in U$). Порядок алфавитов в пометках в обоих полуавтоматах одинаковый. Строим пересечение $(P_B^*)_{\uparrow I} \cap (P_S^*)_{\uparrow U}$.

Шаг 2. Автомат L' имеет то же множество состояний, что и полуавтомат $(P_B^*)_{\uparrow I} \cap (P_S^*)_{\uparrow U}$ в объединении со специальным безразличным состоянием DNC . В автомате L' существует переход $(bs, io, u, b's')$, если и только если в полуавтомате $(P_B^*)_{\uparrow I} \cap (P_S^*)_{\uparrow U}$ есть переход $(bs, iuoy, b's')$ для некоторого $y \in Y$. В автомате L' существует переход (bs, io, u, DNC) , если и только если в состоянии bs под действием входного символа io отсутствует переход в некоторое состояние $b's'$ и в состоянии b автомата B отсутствует переход с входо-выходной парой uoy для всех $y \in Y$. Наибольший полностью определенный подавтомат L автомата L' (если существует) есть наибольшее решение уравнения $X \bullet B \equiv S$.

Теорема 2.11. Если алгоритм 2.5 не возвращает полностью определенный автомат, то уравнение $X \bullet B \equiv S$ для композиции на рис. 2.19

не имеет полностью определенного решения. Пусть L – автомат, построенный по алгоритму 2.5. Полностью определенный детерминированный автомат A является решением уравнения $X \bullet B \equiv S$ (для топологии на рис. 2.19), если и только если A есть редукция автомата L .

□

Таким образом, для композиции на рис. 2.19 можно охарактеризовать все полностью определенные детерминированные решения автоматного уравнения $X \bullet B \equiv S$.

Поскольку для композиций определенного вида сложность решения автоматных уравнений является полиномиальной, то этот факт можно использовать, например, при локальной оптимизации цифровых схем. Из последовательностной схемы выделяется некоторый фрагмент для оптимизации. Далее данный фрагмент представляется последовательной композицией двух последовательностных схем (рис. 1.20). Для каждой из подсхем можно теперь найти общее решение автоматного уравнения, и выбрать в некотором смысле оптимальную редукцию такого решения. Более подробно этот вопрос обсуждается в главе 4.

2.5. Системы синхронных автоматных уравнений

При глобальной оптимизации многокомпонентной сети (раздел 1.6.1) коэффициентом A в уравнении $A \bullet X \equiv S$ является автомат, описывающий совместное поведение всех компонент сети, за исключением оптимизируемой компоненты. При таком глобальном подходе решение автоматного уравнения может оказаться достаточно сложной задачей, если автомат A будет иметь большую размерность. Ввиду ассоциативности операции композиции, можно использовать так называемую локальную оптимизацию (*window approach*). При локальном подходе каждая компонента оптимизируется не относительно всей сети, а только относительно непосредственно связанных с ней компонент. Достоинство такой оптимизации очевидно, так как нет необходимости описывать одним автоматом поведение многокомпонентной сети, т.е. размеры автоматов, коэффициентов уравнения, будут существенно меньше, чем в случае глобального подхода. Несмотря на то, что при локальной оптимизации часть возможных реализаций компоненты будет потеряна, эксперименты показывают, что локальная оптимизация дает хорошие результаты.

2.5.1. Системы синхронных автоматных уравнений при локальной оптимизации автоматных сетей

Рассмотрим сеть из полностью определенных детерминированных автоматов на рис. 2.20, в которой головная компонента последовательно соединена с двумя хвостовыми компонентами.

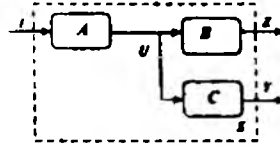


Рис. 2.20. Автоматная сеть из трех компонент

Для локальной оптимизации будем рассматривать фрагменты сети, состоящие из двух компонент: оптимизируемой и соседней компоненты. Для оптимизации компоненты A (рис. 2.20) можно рассмотреть два таких фрагмента: первый фрагмент будет состоять из компонент A и B , второй фрагмент – из компонент A и C (рис. 2.21а). Для каждого фрагмента сети составим соответствующее уравнение, в результате получаем семейство из двух уравнений: $\{X \bullet B \cong S_1; X \bullet C \cong S_2\}$, в которых автоматы S_1 и S_2 описывают поведение последовательных композиций $A \bullet B$ и $A \bullet C$ соответственно. Автомат, являющийся решением одного из уравнений данного множества, может заменить компоненту A без изменения внешнего поведения сети. Таким образом, в качестве оптимальной реализации компоненты A можно выбрать наилучшее решение из множества оптимальных решений этих уравнений, либо оптимальное решение любого из уравнений.

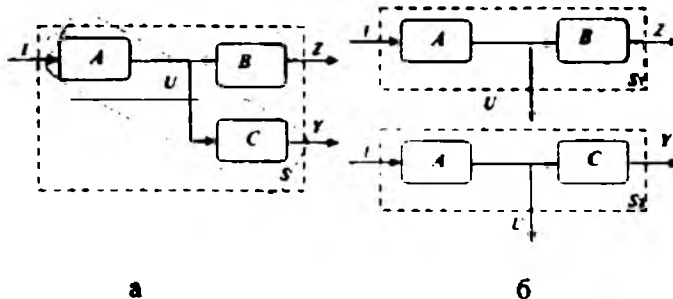


Рис. 2.21, а локальная оптимизация компоненты
б выделение фрагментов

Непосредственной проверкой можно убедиться, что наибольшие решения $Largest_1$ и $Largest_2$ уравнений $X \bullet B \cong S_1, X \bullet C \cong S_2$ эк-

вивалентны автомату A , поскольку все входы и выходы компоненты A для композиции $A \bullet B$ (так же как и для композиции $A \bullet C$) являются внешними полюсами соответствующей автоматной сети (рис. 2.21б). Следовательно, головная компонента автоматной сети на рис. 2.20 не может быть локально оптимизирована посредством решения множества уравнений.

Для оптимизации головной компоненты A необходимо «скрыть» выходной алфавит U компоненты таким образом, чтобы он являлся внутренним алфавитом композиций $A \bullet B$ и $A \bullet C$. Композиция автоматов на рис. 2.20 может быть преобразована следующим образом. Вместо последовательной композиции головной компоненты с двумя хвостовыми компонентами можно рассмотреть композицию из двух последовательных автоматных сетей (рис. 2.22), каждая из которых представляет собой композицию головной компоненты с одной хвостовой компонентой. При таком преобразовании поведение всей системы в целом не изменится.

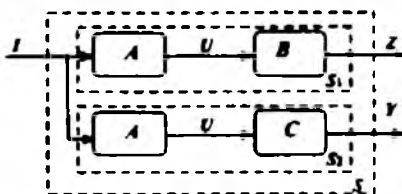


Рис. 2.22. Представление композиции на рис. 2.20 в виде двух схем

Согласно рис. 2.22, при использовании множества уравнений для оптимизации компоненты A , найденное решение должно удовлетворять каждому уравнению из множества. Таким образом, возникает понятие системы автоматных уравнений:

$$\begin{cases} X \bullet B \equiv S_1 \\ X \bullet C \equiv S_2 \end{cases} \quad (2.2)$$

где $S_1 = A \bullet B$ и $S_2 = A \bullet C$. При этом выход U компоненты A является внутренним в композициях $A \bullet B$ и $A \bullet C$.

Теорема 2.12. Автомат *Largest* является решением уравнения $X \bullet (B \bullet C) \equiv S$, если и только если автомат *Largest* является решением системы уравнений (2.2).

□

Таким образом, для локальной оптимизации компоненты на рис. 2.21 нужно решить не одно автоматное уравнение, а систему та-

ких уравнений с одним неизвестным.

Пусть даны полностью определенные наблюдаемые автоматы S_j и A_j , $j = 1, \dots, n$. Системой синхронных автоматных уравнений называется совокупность уравнений вида:

$$\begin{cases} A_1 \bullet X \equiv S_1 \\ A_2 \bullet X \equiv S_2 \\ \dots \\ A_n \bullet X \equiv S_n \end{cases} \quad (2.3)$$

Неизвестная компонента X является автоматом с заданными входным и выходным алфавитами. Автомат B , определенный в алфавитах неизвестной компоненты X , называется *решением системы уравнений* (2.3), если он является решением каждого уравнения системы.

Автомат *Largest* называется *наибольшим решением системы уравнений* (2.3), если любое решение системы уравнений является редукцией автомата *Largest*.

2.5.2. Решение системы синхронных автоматных уравнений

Рассмотрим систему (2.3) синхронных автоматных уравнений

$$\begin{cases} A_1 \bullet X \equiv S_1 \\ A_2 \bullet X \equiv S_2 \\ \dots \\ A_n \bullet X \equiv S_n \end{cases}$$

где S_j и A_j суть полностью определенные наблюдаемые автоматы, $j = 1, \dots, n$.

Без ограничения общности мы полагаем, что каждое уравнение соответствует композиции на рис. 1.10.¹⁵ В этом случае компонента X имеет входной алфавит $I_2 \times U$ и выходной алфавит $O_2 \times V$.

Любое разрешимое автоматное уравнение имеет наибольшее решение *Largest_j*, $j = 1, \dots, n$. Следовательно, решение системы уравнений содержится в каждом наибольшем решении, то есть является редукцией пересечения наибольших решений уравнений системы, и таким образом, для решения системы автоматных уравнений предлагается следующий алгоритм.

Алгоритм 2.6. Нахождение наибольшего решения системы синхронных автоматных уравнений

Вход: Система автоматных уравнений (2.3)

¹⁵ В общем случае топологии композиции для различных автоматов S_j и A_j могут быть различными

Выход: Наибольшее решение системы уравнений (если существует)

Шаг 1. Находим наибольшие решения $Largest_1, \dots, Largest_n$ для всех уравнений системы. Если хотя бы одно уравнение не разрешимо, то система не имеет решения. КОНЕЦ. Иначе строим пересечение $Largest = Largest_1 \cap \dots \cap Largest_n$.

Шаг 2. Если $Largest$ есть решение каждого уравнения, то автомат $Largest$ есть наибольшее решение системы. Если $Largest$ не является решением хотя бы одного уравнения, то система не имеет решения.

□

Пример. Оптимизируем компоненту A , являющуюся головной компонентой автоматной сети на рис. 2.20. На рис. 2.23 представлены диаграммы переходов автоматов A, B и C .

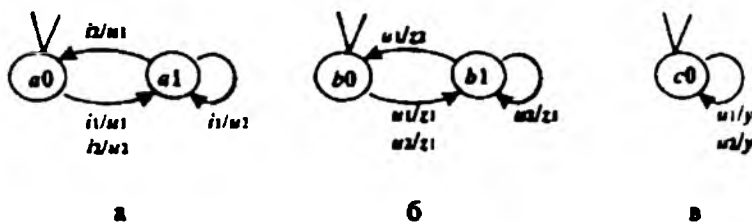


Рис. 2.23, а автомат A
б автомат B
в автомат C

Для оптимизации компоненты A рассмотрим систему уравнений:

$$\begin{cases} X \cdot B \equiv A \cdot B \\ X \cdot C \equiv A \cdot C \end{cases}$$

Наибольшие решения $Largest_1$ и $Largest_2$ уравнений $X \cdot B \equiv A \cdot B$ и $X \cdot C \equiv A \cdot C$ показаны на рис. 2.24.

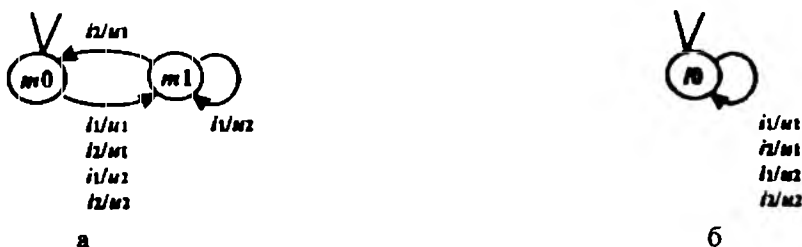


Рис. 2.24, а наибольшее решение $Largest_1$ уравнения $X \cdot B \equiv A \cdot B$
б наибольшее решение $Largest_2$ уравнения $X \cdot C \equiv A \cdot C$

Строим пересечение *Largest* наибольших решений (рис. 2.25а). Автомат *Largest* является решением обеих уравнений системы, то есть *Largest* – наибольшее решение системы уравнений. Более того, поскольку композиция на рис. 2.20 не имеет обратных связей, каждая полностью определенная редукция автомата *Largest* также является решением системы. В качестве оптимального решения можно выбрать редукцию *Lopt* с одним состоянием автомата *Largest* (рис. 2.25б), в то время как исходная головная компонента имела два состояния.

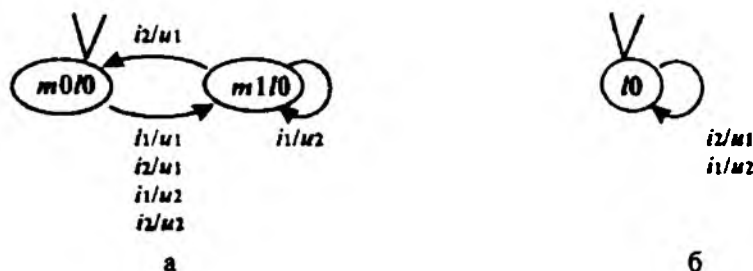


Рис. 2.25, а наибольшее решение *Largest*
 б редукция автомата *Lopt* с одним состоянием

2.5.3. Частные решения системы автоматных уравнений

Так же как для уравнений, для систем уравнений можно рассматривать частные решения.

Алгоритм 2.7. Нахождение наибольшего полностью определенного решения системы синхронных автоматных уравнений

Вход: Система синхронных автоматных уравнений (2.3)

Выход: Наибольшее полностью определенное решение системы уравнений (если существует)

Шаг 1. Находим наибольшие полностью определенные решения L_1, \dots, L_n для всех уравнений системы. Если хотя бы одно уравнение не разрешимо или не имеет полностью определенного решения, то система не имеет полностью определенных решений. **КОНЕЦ.**

Шаг 2. Строим пересечение $M = L_1 \cap \dots \cap L_n$ и находим наибольший полностью определенный подавтомат L автомата M , удаляя итеративно из автомата M состояния, в которых не определен хотя бы один переход и переходы в это состояние. Если автомат M не имеет полностью определенных подавтоматов, то система не имеет полностью определенных решений. **КОНЕЦ**

Шаг 3. Если L есть решение каждого уравнения, то автомат L есть наибольшее полностью определенное решение системы уравнений. Если L не является решением хотя бы одного уравнения, то система уравнений не имеет решения.

□

Наибольшее муравское решение системы синхронных автоматных уравнений (если существует) находится точно так же как наибольшее полностью определенное решение.

Заметим, что если система уравнений имеет прогрессивное решение, т.е. полностью определенное решение, которое является прогрессивным для каждого уравнения, то система имеет и наибольшее прогрессивное решение. Однако алгоритм нахождения наибольшего прогрессивного решения системы (если существует) более сложный, т.к. полученное на шаге 2 пересечение может оказаться решением для каждого уравнения, но не будет являться прогрессивным решением. Заинтересованный читатель может найти необходимые ссылки в списке дополнительной литературы.

2.6. Контрольные вопросы к главе 2

1. Что такое синхронное автоматное уравнение?
2. Какой автомат называется решением автоматного уравнения?
3. Какое решение называется наибольшим решением автоматного уравнения?
4. Какое решение синхронного автоматного уравнения называется полностью определенным решением? Муравским решением? Безопасным решением? Прогрессивным решением?
5. Каким образом можно найти решение автоматного уравнения для хвостовой компоненты последовательной композиции?
6. Каким образом можно найти решение автоматного уравнения для головной компоненты последовательной композиции?
7. Каким образом можно найти решение автоматного уравнения на основе безразличных последовательностей?
8. Как используются полуавтоматы при решении синхронного автоматного уравнения?
9. Как можно найти наибольшее решение синхронного автоматного уравнения в заданных алфавитах без использования полуавтоматов?

10. Как найти наибольшее полностью определенное решение синхронного автоматного уравнения? Каким свойством обладает такое решение?

11. Является ли любая полностью определенная редукция наибольшего решения решением автоматного уравнения?

12. Как найти наибольшее муровское решение синхронного автоматного уравнения? Каким свойством обладает такое решение?

13. Является ли любая полностью определенная редукция наибольшего муровского решения решением автоматного уравнения?

14. Как найти наибольшее решение автоматного уравнения в наибольшем алфавите?

15. Как найти наибольшее решение системы синхронных автоматных уравнений?

3. РЕШЕНИЕ ПАРАЛЛЕЛЬНОГО АВТОМАТНОГО УРАВНЕНИЯ

В этом разделе мы показываем, каким образом можно решать уравнение для параллельной композиции автоматов. Мы рассматриваем автоматную сеть с обратными связями, поскольку для последовательной автоматной сети решение параллельного автоматного уравнения будет таким же, как для синхронного случая. Так же как и в предыдущей главе, мы рассматриваем достаточно общую структуру сети (рис. 1.10); однако в этой главе компоненты будут функционировать по очереди (асинхронно). Компонента начинает функционировать, получив входной сигнал от другой компоненты или из внешней среды. Компоненты функционируют в режиме диалога, и следующий внешний сигнал может поступить на вход сети только после того, как между компонентами закончился внутренний диалог. Если обе компоненты композиции суть полностью определенные автоматы, то внешний входной сигнал подается только тогда, когда одна из компонент выдала внешний выходной сигнал. К сожалению, существуют сети, в которых при подаче внешней входной последовательности компоненты начинают бесконечный внутренний диалог. В этом случае мы говорим, что композиция допускает осцилляцию. Если компоненты являются частичными автоматами, то в сети возможны и тупиковые ситуации, когда поведение компоненты не определено под действием поступившего внутреннего входного сигнала. В данном пособии мы не рассматриваем возникновение тупиковых ситуаций, полагая, что компоненты являются полностью определенными автоматами.

3.1. Решение параллельного автоматного уравнения

Как и в предыдущей главе, мы полагаем, что все автоматы в уравнении $A \diamond X \equiv S$ (неравенстве $A \diamond X \leq S$) являются полностью определенными автоматами. При решении уравнений для частичных автоматов можно воспользоваться теми же литературными источниками, что и при решении синхронных автоматных уравнений.

Наибольшее решение автоматного уравнения можно получить, выполняя операции над языками, реализуемыми автоматами-коэффициентами уравнения. Рассмотрим решение автоматного уравнения $A \diamond X \equiv S$ для автоматной сети на рис. 1.10. Поскольку операция параллельной композиции автоматов определяется через соответ-

вующую композицию их языков, то для решения автоматного неравенства $A \diamond X \leq S$ достаточно найти решение языкового неравенства $((L_A^{\cup})_{\Pi_2 \cup O_2} \cap (L_X^{\cup})_{\Pi_1 \cup O_1})_{\Pi_1 \cup \Pi_2 \cup O_1 \cup O_2} \subseteq L_S^{\cup}$. В последнем неравенстве язык L_X^{\cup} определен в алфавите $I_2 \cup O_2 \cup U \cup V$. Иными словами, в языке, содержащем все конечные последовательности над алфавитом $I_2 \cup O_2 \cup U \cup V$, нужно оставить только такие последовательности, которые в параллельной композиции с языком автомата A реализуют входо-выходные последовательности автомата S . Соответственно, последовательность $\alpha \in (I_2 \cup O_2 \cup U \cup V)^*$ может принадлежать решению неравенства $((L_A^{\cup})_{\Pi_2 \cup O_2} \cap (L_X^{\cup})_{\Pi_1 \cup O_1})_{\Pi_1 \cup \Pi_2 \cup O_1 \cup O_2} \subseteq L_S^{\cup}$, если и только если $(L_A^{\cup})_{\Pi_2 \cup O_2} \cap \alpha_{\Pi_1 \cup O_1})_{\Pi_1 \cup \Pi_2 \cup O_1 \cup O_2}$ не пересекает дополнение $\overline{L_S^{\cup}}$ языка L_S^{\cup} , т.е.

$$(L_A^{\cup})_{\Pi_2 \cup O_2} \cap \alpha_{\Pi_1 \cup O_1})_{\Pi_1 \cup \Pi_2 \cup O_1 \cup O_2} \cap \overline{L_S^{\cup}} = \emptyset. \quad (3.1)$$

Выражение (3.1) эквивалентно выражению

$$\alpha \in ((L_A^{\cup})_{\Pi_2 \cup O_2} \cap \overline{(L_S^{\cup})_{\Pi_1 \cup O_1}})_{\Pi_2 \cup O_2 \cup U \cup V}$$

или, что то же самое,

$$\alpha \in \overline{\overline{((L_A^{\cup})_{\Pi_2 \cup O_2} \cap (L_S^{\cup})_{\Pi_1 \cup O_1})_{\Pi_2 \cup O_2 \cup U \cup V}}} = \overline{L_A^{\cup} \diamond L_S^{\cup}}$$

Таким образом, язык $\overline{L_A^{\cup} \diamond L_S^{\cup}}$ есть наибольшее решение неравенства $((L_A^{\cup})_{\Pi_2 \cup O_2} \cap (L_X^{\cup})_{\Pi_1 \cup O_1})_{\Pi_1 \cup \Pi_2 \cup O_1 \cup O_2} \subseteq L_S^{\cup}$. Т.к. решение неравенства должно быть языком конечного автомата с входным алфавитом $I_2 \cup U$ и выходным алфавитом $O_2 \cup V$, то полученный язык $\overline{L_A^{\cup} \diamond L_S^{\cup}}$ необходимо пересечь с множеством $\overline{((I_2 \cup U)(O_2 \cup V))^*}$ и выделить максимальное подмножество $\overline{[L_A^{\cup} \diamond L_S^{\cup} \cap ((I_2 \cup U)(O_2 \cup V))^*]^{IO-pref}}$, т.е. подмножество, которое замкнуто относительно всех начальных отрезков с финальными символами $b \in O_2 \cup V$. Иными словами, вместе с каждой последовательностью $a_1 b_1 \dots a_k b_k$, $a_j \in I_2 \cup U$, $b_j \in O_2 \cup V$, $j = 1, \dots, k$, наибольшему решению автоматного уравнения должен принадлежать ее любой начальный отрезок $a_1 b_1 \dots a_n b_n$, $n < k$. Соответственно, автомат с языком $\overline{[L_A^{\cup} \diamond L_S^{\cup} \cap ((I_2 \cup U)(O_2 \cup V))^*]^{IO-pref}}$ есть наибольшее ре-

шение¹⁶ автоматного неравенства $A \diamond X \leq S$ ¹⁷, и для автоматного неравенства $A \diamond X \leq S$, соответствующего автоматной сети на рис. 1.10, справедливо следующее утверждение.

Теорема 3.1. Наибольшим решением автоматного неравенства $A \diamond X \leq S$ является автомат *Largest* с языком $\{\underline{L_1^* \diamond L_2^*} \cap ((I_2 \cup U)(O_2 \cup V))^*\}$ ^{IO-pref}. Более того, любая редукция автомата *Largest* также является решением автоматного неравенства $A \diamond X \leq S$.

□

Обычно в качестве наибольшего решения выбирается наблюдаемый приведенный автомат, хотя выбор конкретного автомата для представления наибольшего решения автоматного уравнения существенно зависит от дальнейшего использования такого решения.

Заметим, что, как показывает следующий пример, в отличие от синхронного автоматного неравенства, параллельное автоматное неравенство $A \diamond X \leq S$ не всегда разрешимо. Причиной является тот факт, что распространение языка тривиального автомата (автомата с пустым отношением переходов) содержит не только пустую последовательность, и, соответственно, может оказаться, что тривиальный автомат в композиции с некоторым автоматом A не является редукцией автомата S .

Пример. Чтобы проиллюстрировать на примере, что неравенство $A \diamond X \leq S$ не всегда имеет решение, обратимся к рис. 3.1.

¹⁶ Подобно синхронному автоматному неравенству, существует не единственное наибольшее решение неравенства $A \diamond X \leq S$. Однако все эти решения эквивалентны, т.к. реализуют один и тот же язык $\underline{L_1^* \diamond L_2^*} \cap ((I_2 \cup U)(O_2 \cup V))^*$.

¹⁷ К сожалению, выделение наибольшего IO-префикс замкнутого подмножества из языка $\underline{L_1^* \diamond L_2^*} \cap ((I_2 \cup U)(O_2 \cup V))^*$ является необходимым шагом при решении уравнений для конечных автоматов. Можно привести примеры, когда язык $\underline{L_1^* \diamond L_2^*} \cap ((I_2 \cup U)(O_2 \cup V))^*$ не обладает этим свойством.

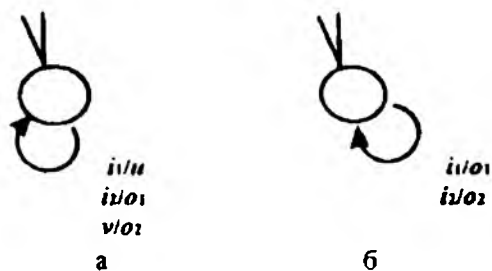


Рис 3.1, а автомат А
б автомат S

На рис. 3.1 показаны диаграммы переходов автомата А (рис. 3.1а) с входным алфавитом $I \cup V$, $I = \{i_1, i_2\}$, $V = \{v\}$, и выходным алфавитом $O \cup U$, $O = \{o_1, o_2\}$, $U = \{u\}$, и автомата S (рис. 3.1б) с входным алфавитом $\{i_1, i_2\}$ и выходным алфавитом $\{o_1, o_2\}$ для автоматной сети на рис. 1.16. Наибольшее решение языкового неравенства $((L_A)_{\Pi \cup O_2} \cap (L_X)_{\Pi \cup O_1})_{\Pi \cup O_2} \subseteq L_S$ в алфавите $\{v, u\}$ есть пустой язык, т.к. композиция автомата А и тривиального автомата, язык которого содержит только пустое слово, не является редукцией автомата S. На входной символ i_2 композиция отвечает выходным символом o_1 , в то время как в автомате S выходным символом является o_2 . Таким образом, параллельная композиция автомата А с любым автоматом В с входным алфавитом $\{u\}$ и выходным алфавитом $\{v\}$ не является редукцией автомата S.

Поскольку языки, реализуемые автоматами, являются регулярными, то при построении наибольшего решения автоматного неравенства $A \diamond X \leq S$ все действия можно производить над соответствующими полуавтоматами. Ниже мы приводим алгоритм построения наибольшего решения параллельного автоматного неравенства.

Алгоритм 3.1. Нахождение наибольшего решения параллельного автоматного неравенства

Вход: Автоматы $A = (A, I_1 \cup V, O_1 \cup U, T_A, a_0)$ и $S = (S, I_2 \cup V, O_2 \cup U, T_S, s_0)$

Выход: Наибольшее решение *Largest* автоматного неравенства $A \diamond X \leq S$

Шаг 1. По автоматам А и S строим соответствующие полуавтоматы P_A^\cup и P_S^\cup .

Шаг 2. Строим полуавтомат $\overline{P_S^\cup}$: неопределенные переходы доопределяются переходом в безразличное состояние DNC.

Шаг 3. Строим полуавтоматы $(P_A^\vee)_{\Pi_{I_2 \cup O_2}}$ и $(P_S^\vee)_{\Pi_{U \cup V}}$, добавляя в каждом состоянии петлю, помеченную всеми символами из алфавита $I_2 \cup O_2$ (соответственно, символами из алфавита $U \cup V$).

Шаг 4. Строим пересечение полуавтоматов $(P_A^\vee)_{\Pi_{I_2 \cup O_2}} \cap \overline{(P_S^\vee)_{\Pi_{U \cup V}}}$.

Шаг 5. Находим проекцию полуавтомата $(P_A^\vee)_{\Pi_{I_2 \cup O_2}} \cap \overline{(P_S^\vee)_{\Pi_{U \cup V}}}$ на алфавиты решения $I_2 \cup U \cup O_2 \cup V$, заменяя на каждом переходе пометку a пустым словом ϵ , если $a \in I_2 \cup U \cup O_2 \cup V$. Детерминизируем полученный полуавтомат, находим его дополнение и пересекаем полученное дополнение с полуавтоматом, который представляет язык $((I_2 \cup U)(O_2 \cup V))^*$. Из полученного полуавтомата удаляем все нефинальные состояния, в которые есть переход под действием символов из алфавита $O_2 \cup V$, вместе с переходами из этих состояний и в эти состояния, и получаем полуавтомат N .

Шаг 6. Множество финальных состояний полуавтомата N объявляем множеством состояний автомата *Largest*. В автомате *Largest* есть переход (m, i_2u, o_2v, m') , если и только если в полуавтомате N существует нефинальное состояние m'' и пара последовательных переходов (m, i_2u, m'') и (m'', o_2v, m') . Автомат *Largest* с входным алфавитом $I_2 \cup U$ и выходным алфавитом $O_2 \cup V$ есть наибольшее решение неравенства $A \diamond X \leq S$.

□

Так как *Largest* – наибольшее решение неравенства $A \diamond X \leq S$, то наибольшее решение уравнения $A \diamond X \equiv S$ совпадает с *Largest*, если и только если композиция $A \diamond Largest$ эквивалентна автомату S .

Теорема 3.2. Пусть *Largest* – наибольшее решение неравенства $A \diamond X \leq S$. Если композиция $A \diamond Largest$ эквивалентна автомату S , то *Largest* – наибольшее решение уравнения $A \diamond X \equiv S$. Если композиция $A \diamond Largest$ не эквивалентна автомату S , то автоматное уравнение $A \diamond X \equiv S$ неразрешимо.

□

Пример. В качестве примера проверим, насколько эффективно реализована кофейная машина в кофейне (рис. 1.4). Соответствующий официанту полуавтомат P_A^\vee и полуавтомат $\overline{P_S^\vee} \Pi_{\{C, B, D, E\}}$, где $\overline{P_S^\vee}$ есть дополнение полуавтомата кофейни, приведены на рис. 3.2а и 3.2б. На

рис. 3.2в показан полуавтомат $(P_A^\infty \cap (P_S^\infty)_{\{C, B, L, E\}})$. Пересечение этого полуавтомата с полуавтоматом, реализующим язык $(\{C, B\} \{L, E\})^*$ и имеющим состояния a и b , показано на рис. 3.2г. Удаляем все нефинальные состояния, в которые есть переходы под действием символов L и E (в данном случае таких состояний нет), и строим соответствующий автомат с входным алфавитом $\{C, B\}$ и выходным алфавитом $\{L, E\}$. Полученный автомат (рис. 3.2д) есть наибольшее решение уравнения $Waiter \diamond X \equiv Coffee-shop$. Непосредственной проверкой можно убедиться, что кофейная машина с одним состоянием (рис. 1.19а) является редукцией автомата на рис. 3.2д.

Точно так же, как для синхронных автоматных уравнений, сложность решения произвольного параллельного автоматного уравнения (алгоритм 3.1) является экспоненциальной, т.к. на шаге 5 используется детерминизация полуавтомата. Кроме того, известно, что в общем случае не всякая даже полностью определенная редукция наибольшего решения является решением автоматного уравнения $A \diamond X \equiv S$, т.к. компоненты могут начать бесконечный диалог (наличие осцилляции), и в данный момент не известно эффективного метода характеристики всех решений автоматного уравнения. Точно так же, как и в предыдущем разделе, можно найти часть наибольшего решения на основе использования безразличных последовательностей, поэтому разработка такого алгоритма предлагается в качестве самостоятельного упражнения. В следующих разделах мы рассматриваем частные решения параллельного автоматного уравнения, которые не инициируют осцилляций в композиции с автоматом A .

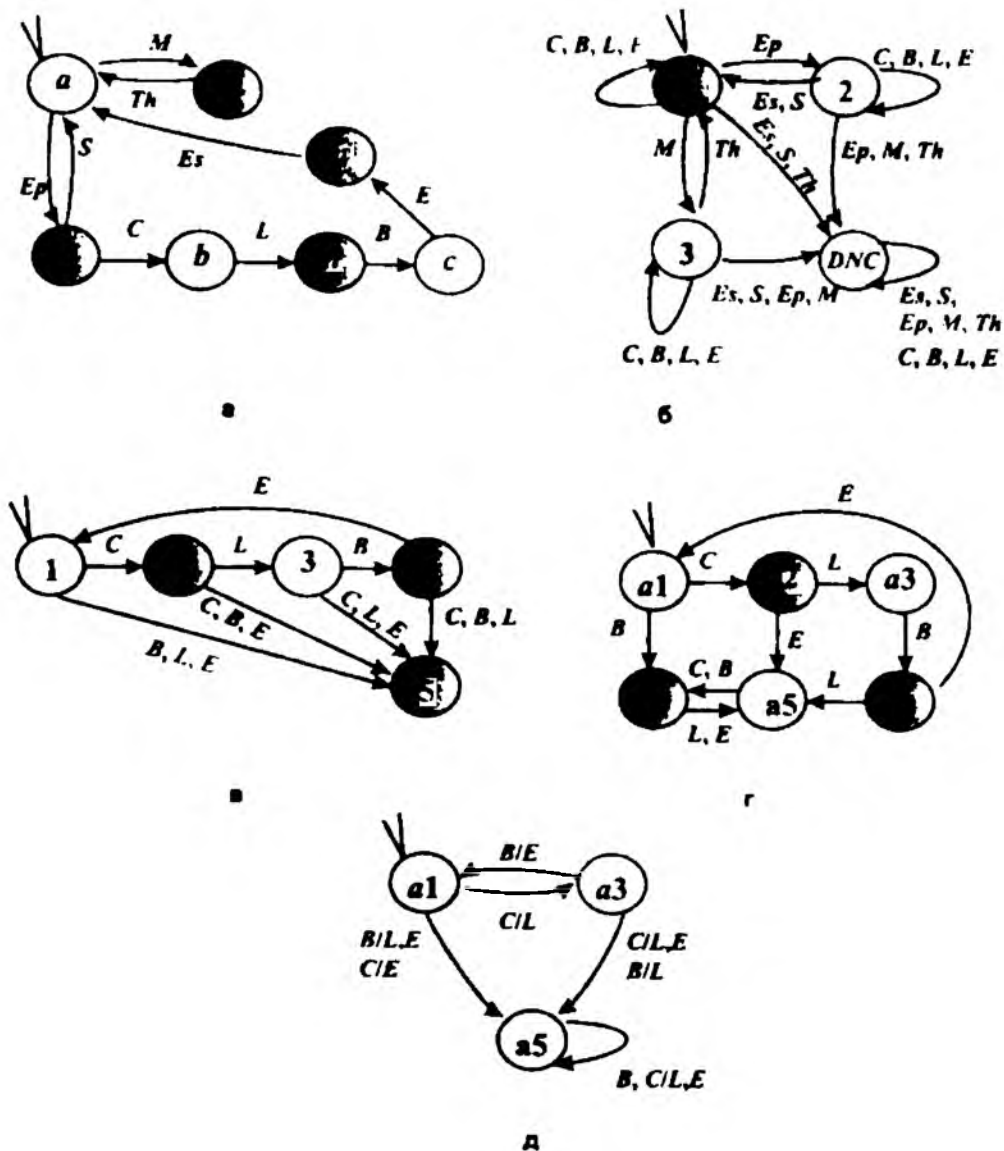


Рис. 3.2, а полуавтомат P_A

б полуавтомат $(P_S)_{(C,B,L,E)}$

в полуавтомат $(P_A \cap (P_S)_{(C,B,L,E)})$

г пересечение полуавтомата $(P_A \cap ((P_S)_{(C,B,L,E)} \cap ((C, B) \{L, E\})^*))$

с полуавтоматом, реализующим язык $((C, B) \{L, E\})^*$
 д наибольшее решение уравнения $Waiter \diamond X \cong Coffee-shop$

Задачи для самостоятельного решения

1. Найти наибольшее решение параллельного автоматного уравнения $A \diamond X \cong S$ с входным алфавитом $\{u\}$ и выходным алфавитом $\{v_1, v_2\}$ для автоматной сети на рис. 1.16 и автоматов на рис. 3.3.

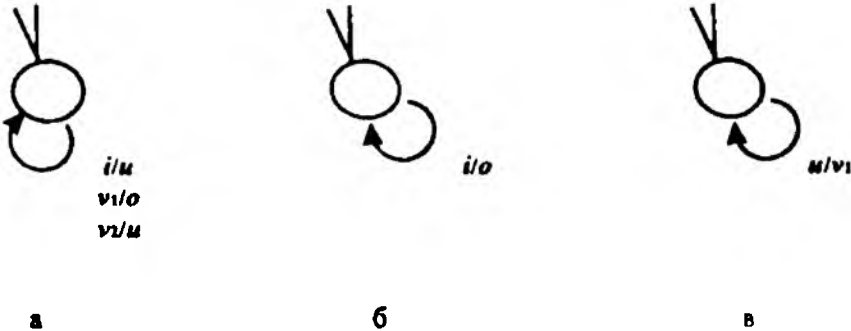


Рис. 3.3, а автомат A с входным алфавитом $\{i, v_1, v_2\}$ и выходным алфавитом $\{u, o\}$
б автомат S с входным алфавитом $\{i\}$ и выходным алфавитом $\{o\}$
в одно из решений уравнения $A \diamond X \cong S$ с входным алфавитом $\{u\}$ и выходным алфавитом $\{v_1, v_2\}$

2. Найти в наибольшем решении в упражнении 1 два частных решения уравнения (если существуют). (Указание: необходимо убедиться, что выбранные редукции наибольшего решения являются решениями уравнения, т.е. в композиции с автоматом A эквивалентны автомату S).
3. Решить уравнение в упражнении 1 на основе безразличных последовательностей с использованием неизвестной компоненты на рис. 3.3в. Проверить, можно ли таким способом найти все решения уравнения.
- 4*. Разработать алгоритм нахождения части наибольшего решения параллельного автоматного уравнения на основе безразличных последовательностей.

3.2. l -ограниченные решения

Как уже отмечалось, с практической точки зрения наиболее интересны полностью определенные решения автоматного уравнения $A \diamond X \cong S$, причем такие, что композиция решения с автоматом A не допускает осцилляций. Наибольшее полностью определенное решение параллельного автоматного уравнения, как и в случае синхронного

уравнения, находится так же как наибольший полностью определенный подавтомат наибольшего решения уравнения. Однако не всякая полностью определенная редукция наибольшего полностью определенного решения является решением параллельного автоматного уравнения, поскольку в общем случае в композиции автомата A с полностью определенным решением уравнения возможна осцилляция. В этом разделе для заданного натурального числа l мы рассматриваем специальное так называемое l -ограниченное решение параллельного автоматного уравнения, которое в некотором смысле является аналогом муровского решения для синхронного автоматного уравнения. Во-первых, композиция автомата A с любым полностью определенным l -ограниченным решением B не допускает осцилляции; более того, если автомат A полностью определенный, то и композиция $A \diamond B$ будет полностью определенным автоматом.

Мы говорим, что решение B автоматного уравнения $A \diamond X \equiv S$ (или автоматного неравенства $A \diamond X \leq S$) является l -ограниченным, если при подаче любого внешнего сигнала на вход автоматной сети компоненты A и B обмениваются не более чем l внутренними сигналами до того, как сеть произведет внешний выходной сигнал. Для того чтобы описать все такие решения, мы вводим понятие l -ограниченной композиции языков.

Пусть языки L_1 и L_2 определены в алфавитах $I \cup O$ и $U \cup O$ соответственно, причем алфавиты I , O , U попарно не пересекаются. Тогда под l -ограниченной композицией $L_1 \diamond_l L_2$ языков L_1 и L_2 понимается язык $[(L_1)_{\Pi O} \cap (L_2)_{\Pi U} \cap (I \cup O)^*_{\Pi(U, O)}]_{\Pi(U, O)}$. Для языка L в алфавите I операция $L_{\Pi(U, l)}$ отличается от операции U -распространения только тем, что между буквами слов языка L разрешается вставлять не любое слово в алфавите U , а только слова, длина которых не превышает l . Если $l = \infty$, то l -ограниченная композиция языков L_1 и L_2 совпадает с параллельной композицией этих языков.

Пример. Рассмотрим язык L , состоящий из одного слова $i_1 i_2$. Пусть $l = 1$ и $U = \{u\}$. Тогда $L_{\Pi(U, 1)} = \{i_1 u i_2, i_1 u i_2, i_1 u i_2, u i_1 u i_2, u i_1 u i_2, i_1 u i_2 u, u i_1 u i_2 u\}$.

Известно, что для разрешимого языкового неравенства $L \diamond_l L_X \subseteq N$ существует наибольшее решение, и так же, как в предыдущих разделах, мы будем искать l -ограниченное решение автоматного уравнения (неравенства) на основе решения соответствующего уравнения (неравенства) для автоматных языков. Поскольку автоматные языки явля-

ются регулярными, то мы сначала показываем, как построить полуавтомат, представляющий l -распространение $L_{\Pi U, l}$ регулярного языка L .

Для заданного полуавтомата $P = (P, I, T_P, F_P, p_0)$ с языком L в алфавите I , можно построить полуавтомат $A_{\Pi U, l}$, представляющий l -распространение $L_{\Pi U, l}$ языка L на алфавит $U, I \cap U = \emptyset$, по следующим правилам.

1. Множество состояний полуавтомата $A_{\Pi U, l}$ есть $P \cup \{(p, j) : p \in P, 1 \leq j \leq l\}$.
2. Отношение переходов полуавтомата $A_{\Pi U, l}$ есть:
 $\delta_P \cup \{(u, p, (p, 1) : u \in U, p \in P\} \cup \{(u, (p, j), (p, j+1) : u \in U, p \in P, 1 \leq j \leq l\} \cup \{(i, (p, j), p') : (i, p, p') \in \delta_P, 1 \leq j \leq l\}$.
3. Начальным состоянием полуавтомата $A_{\Pi U, l}$ является состояние p_0 ; финальные состояния полуавтомата $A_{\Pi U, l}$ суть состояния полуавтомата P и все состояния (p, j) , где $1 \leq j \leq l$ и $p \in F_P$.

Пример. Для языка L из предыдущего примера соответствующий полуавтомат приведен на рис. 3.4а. На рис. 3.4б приведен полуавтомат, представляющий язык $L_{\Pi U, 1}$.

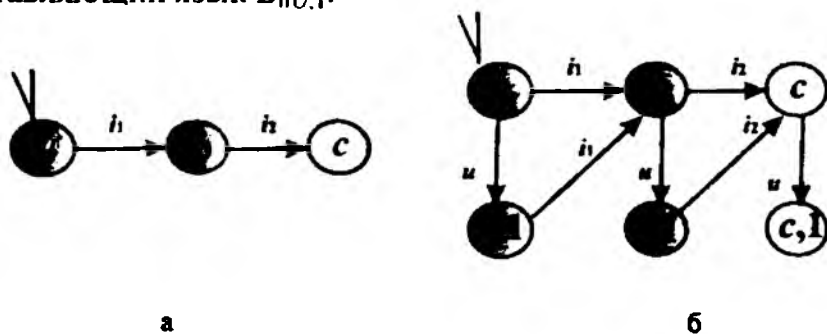


Рис. 3.4, а полуавтомат, представляющий язык $L = \{i_1 i_2\}$
 б полуавтомат, представляющий язык $L_{\Pi U, 1}$

Соответственно, параллельная l -ограниченная композиция $A \diamond_l B$ автоматов A и B (рис. 1.10) есть приведенный наблюдаемый автомат с языком

$$L(A \diamond_l B) = ((L_A^c)_{\Pi U, \infty} \cap (L_B^c)_{\Pi U, \infty}) \cap (I \cup O)^*_{\Pi U, l} \cap (IO)^*$$

Если $l = \infty$, то l -ограниченная композиция $A \diamond_l B$ совпадает с параллельной композицией автоматов A и B .

Можно показать, что решение B автоматного уравнения $A \diamond X \equiv S$ (автоматного неравенства $A \diamond X \leq S$) является l -ограниченным, если B есть решение уравнения $A \diamond_l X \equiv S$ (или, соответственно, автоматного

неравенства $A \diamond X \leq S$); такое наибольшее решение уравнения (или неравенства) будет *наибольшим* l -ограниченным решением.

Любое полностью определенное l -ограниченное решение, вообще говоря, не инициирует осцилляций в композиции с полностью определенным автоматом A , поскольку для любого входного символа обязательно будет произведен внешний выходной сигнал, после того, как компоненты обменяются не более чем l внутренними символами. Таким образом, если A является полностью определенным автоматом и B – полностью определенное l -ограниченное решение автоматного неравенства $A \diamond X \leq S$, то композиция $A \diamond B$ также является полностью определенным автоматом.

Теорема 3.3. Пусть в автоматном неравенстве $A \diamond X \leq S$ автомат A является полностью определенным и B – полностью определенное l -ограниченное решение автоматного неравенства. Тогда композиция $A \diamond B$ также является полностью определенным автоматом. □

Следствие. Пусть в автоматном уравнении $A \diamond X \equiv S$ автомат A является полностью определенным, автомат S является детерминированным и B – полностью определенное l -ограниченное решение автоматного уравнения. Тогда любая полностью определенная редукция автомата B также является решением уравнения $A \diamond X \equiv S$. □

Таким образом, в случае, когда автоматы A и S полностью определенные и детерминированные, наибольшее полностью определенное l -ограниченное решение M уравнения $A \diamond X \equiv S$ характеризует семейство решений уравнения, а именно, все полностью определенные редукции автомата M также являются решениями уравнения.

Наибольшее l -ограниченное решение параллельного автоматного уравнения можно получить, решая соответствующее языковое неравенство $L_A \diamond_l L_X \subseteq L_S$. Соответственно, наибольшее l -ограниченное решение автоматного уравнения есть автомат с языком

$$\overline{[(L_A)_{\cap I_1 \cup O_2} \cap (L_S \cap (IO)^*)_{\cap (U, I)}]_{\cup I_2 \cup O_1 \cup V} \cap ((I_2 \cup U)(O_2 \cup V))^*]^{IO\text{-pref}}}$$

Алгоритм 3.2. Нахождение наибольшего l -ограниченного решения параллельного автоматного неравенства

Вход: Автоматы $A = (A, I_1 \cup V, U \cup O_1, T_A, a_0)$ и $S = (S, I_1 \cup I_2, O_1 \cup O_2, T_S, s_0)$, натуральное число $l \geq 1$

Выход: Наибольшее l -ограниченное решение M автоматного неравенства $A \diamond X \leq S$

Шаг 1. По автоматам A и S строим соответствующие полуавтоматы P_A^\cup и P_S^\cup .

Шаг 2. Строим полуавтомат $\overline{P_S^\cup}$: неопределенные переходы доопределяются переходами в безразличное состояние DNC .

Шаг 3. Строим полуавтомат $(P_A^\cup)_{\Pi_{I_2 \cup O_2}}$, добавляя в каждом состоянии полуавтомата P_A^\cup петлю, помеченную всеми символами из алфавита $I_2 \cup O_2$. Обозначим через K пересечение полуавтомата $\overline{P_S^\cup}$ с полуавтоматом, представляющим язык $(IO)^{18}$. Полуавтомат $K_{\Pi_{U \cup V}}$ строится на основе описанной выше процедуры.

Шаг 4. Строим пересечение полуавтоматов $(P_A^\cup)_{\Pi_{I_2 \cup O_2}} \cap K_{\Pi_{U \cup V}}$.

Шаг 5. Находим проекцию полуавтомата $(P_A^\cup)_{\Pi_{I_2 \cup O_2}} \cap (\overline{P_S^\cup} \cap (IO)^*)_{\Pi_{U \cup V}}$ на алфавиты решения $I_2 \cup O_2 \cup U \cup V$, заменяя на каждом переходе пометку a пустым словом ϵ , если $a \notin I_2 \cup O_2 \cup U \cup V$. Детерминизируем полученный полуавтомат, находим его дополнение и пересекаем полученное дополнение с полуавтоматом, который представляет язык $((I_2 \cup U)(O_2 \cup V))^*$. Из полученного полуавтомата удаляем все нефинальные состояния, в которые есть переход под действием символов из алфавита $O_2 \cup V$, вместе с переходами из этих состояний и в эти состояния, и получаем полуавтомат N .

Шаг 6. Множество финальных состояний полуавтомата N объявляем множеством состояний автомата M . В автомате M есть переход (m, i_2u, o_2v, m') , если и только если в полуавтомате N существует нефинальное состояние m'' и пара последовательных переходов (m, i_2u, m'') и (m'', o_2v, m') . Автомат M с входным алфавитом $I_2 \cup U$ и выходным алфавитом $O_2 \cup V$ есть наибольшее l -ограниченное решение неравенства $A \diamond_l X \leq S$.

□

Так как M – наибольшее l -ограниченное решение неравенства $A \diamond_l X \leq S$, то M является наибольшим l -ограниченным решением урав-

¹⁸ $I = I_1 \cup I_2, O = O_1 \cup O_2$

нения $A \diamond X \equiv S$, если и только если композиция $A \diamond M$ эквивалентна автомату S .

Теорема 3.4. Пусть M – автомат с языком

$$\begin{aligned} & [((L_A)_{\Pi_1 \cup O_2} \cap (L_S \cap (IO)^*)_{\Pi_1 \cup O_2})_{\Pi_1 \cup O_2 \cup U \cup V} \cap \\ & ((I_2 \cup U)(O_2 \cup V)^*)^{IO\text{-pref}}. \end{aligned}$$

1. M есть наибольшее l -ограниченное решение автоматного неравенства $A \diamond X \leq S$.

2. Если композиция $A \diamond M \equiv S$, то M – наибольшее l -ограниченное решение автоматного уравнения $A \diamond X \equiv S$. Иначе автоматное уравнение $A \diamond X \equiv S$ не имеет l -ограниченного решения.

3. Если M – l -ограниченное решение автоматного уравнения $A \diamond X \equiv S$, A – полностью определенный автомат и B – полностью определенная редукция автомата M , то композиция $A \diamond B$ есть полностью определенный автомат.

4. Если S – детерминированный полностью определенный автомат, и M – наибольшее l -ограниченное решение автоматного уравнения $A \diamond X \equiv S$, то полностью определенный автомат B является решением автоматного уравнения $A \diamond B \equiv S$, если и только если B есть редукция автомата M .

□

Таким образом, согласно теореме 3.4, l -ограниченные решения автоматного уравнения привлекательны с практической точки зрения. Однако следует отметить и в некотором смысле отрицательные свойства таких решений.

1. Наибольшее решение и наибольшее l -ограниченное решение неравенства $A \diamond X \leq S$ (уравнения $A \diamond X \equiv S$) в общем случае не совпадают. Наибольшее l -ограниченное решение всегда есть редукция наибольшего решения неравенства $A \diamond X \leq S$.

2. Если уравнение $A \diamond_l X \equiv S$ не является разрешимым, то уравнение $A \diamond X \equiv S$ может быть разрешимо. Просто для любого решения будет существовать внешний входной символ, для которого внешний выходной символ будет получен после обмена более чем l внутренними (невидимыми для внешнего наблюдателя) действиями. Соответственно возможно, что при $n > l$ для данного уравнения существует n -ограниченное решение. Если при любом l разрешимое уравнение $A \diamond_l X \equiv S$ не имеет l -ограниченного решения, то композиция автомата A с любым решением уравнения допускает осцилляцию.

Задачи для самостоятельного решения

1. Для полуавтомата P на рис. 3.5 построить полуавтомат $A_{U,l}$, где $U = \{u_1, u_2\}$ и $l = 2$.

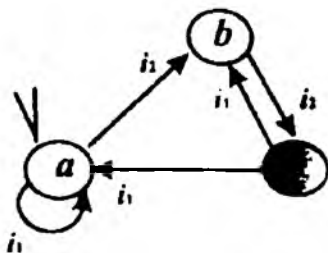


Рис 3.5 Полуавтомат P

2. Проверить, имеет ли уравнение для автоматов на рис. 3.3 наибольшее 2-ограниченное решение.

3. Найти наибольшее 2-ограниченное решение для кофейной машины (рис. 1.46). Если такое решение существует, то взять любую полностью определенную редукцию наибольшего 2-ограниченного решения и убедиться, что такая редукция является решением уравнения.

3.3. Прогрессивные решения параллельного автоматного уравнения

В данном разделе мы вводим понятие (полностью определенного) прогрессивного решения, которое также не инициирует осцилляций в композиции и является в некотором смысле более широким, чем l -ограниченное решение, т.к. включает все l -ограниченные решения. Известно, что если автоматное уравнение имеет прогрессивное решение, то уравнение имеет и наибольшее прогрессивное решение. Однако построение наибольшего прогрессивного решения намного сложнее, чем построение наибольшего полностью определенного решения, наибольшего l -ограниченного решения. Известно, что в общем случае наибольшее прогрессивное решение является не подавтоматом, а только редукцией наибольшего решения уравнения, т.е. не может быть получено из наибольшего решения путем итеративного удаления «плохих» состояний. Поэтому мы не приводим алгоритм нахождения наибольшего прогрессивного решения. Заинтересованный читатель

может найти необходимые ссылки в списке дополнительной литературы.

Решение *SOI* автоматного уравнения $A \diamond X \leq S$ называется *прогрессивным*, если *SOI* - полностью определенный автомат, и для любого слова α из языка $Init(L_A) \cap Init(L_{SOI}) \cap Init(IO)^*$, где $Init(L)$ содержит все начальные отрезки слов из L , существует продолжение в языке $(L_A) \cap (L_{SOI}) \cap (IO)^*$. Таким образом, решение *SOI* будет прогрессивным, если из каждого состояния полуавтомата $(P_A) \cap (P_{SOI})$ в пересечении с полуавтоматом, представляющим язык $(IO)^*$, достижимо финальное состояние, и в каждом финальном состоянии существует переход по каждому входному символу из алфавита $I = I_1 \cup I_2$.

Теорема 3.5. Для любого $l > 0$ любое полностью определенное l -ограниченное решение автоматного уравнения $A \diamond X \equiv S$ является прогрессивным. □

Заметим, что наибольшее прогрессивное решение частично характеризует «хорошие» решения автоматного уравнения, т.е. решения, в композиции которых с автоматом (известной часть системы) отсутствует осцилляция. В частности, если уравнение не имеет наибольшего прогрессивного решения, то такое уравнение вообще не имеет прогрессивных решений. Однако, как показывает следующий пример, для полной характеристики «хороших» (с отсутствием осцилляции) решений наибольшего прогрессивного решения не достаточно.

Пример. Рассмотрим уравнение $A \diamond X \leq S$ для автоматов, представленных на рис. 3.6а и 3.6б. Наибольшее прогрессивное решение *SOI* показано на рис. 3.6в. Непосредственной проверкой нетрудно убедиться, что полностью определенная редукция R наибольшего прогрессивного решения, показанная на рис. 3.6г, не является прогрессивным решением уравнения. Полуавтомат, представляющий язык композиции наибольшего прогрессивного решения и автомата A , показан на рис. 3.6д; полуавтомат, представляющий язык композиции автомата A и редукции R , показан на рис. 3.6е. Однако, если удалить из наибольшего решения переход под действием входо-выходной пары u/v_2 , то часть прогрессивных решений уравнения будет потеряна.

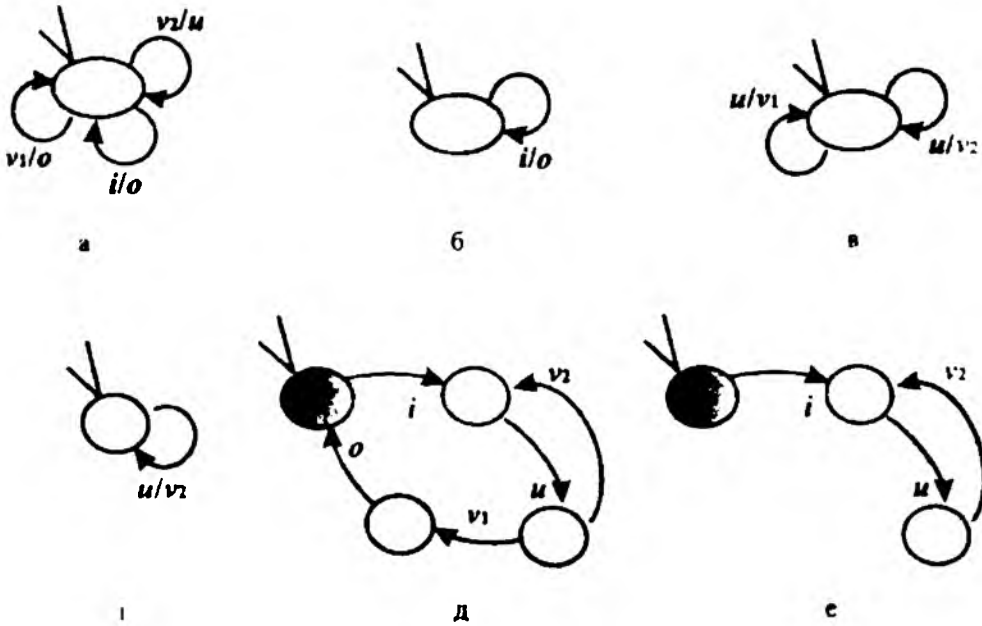


Рис. 3.6, а автомат A
 б автомат S
 в наибольшее прогрессивное решение S_0
 неравенства $A \diamond X \leq S$
 г полностью определенная редукция R наи-
 большего прогрессивного решения, кото-
 рая не является решением
 д полуавтомат, представляющий язык
 композиции автоматов A и S_0
 е полуавтомат, представляющий язык
 композиции автоматов A и R

В списке дополнительной литературы заинтересованный читатель может найти ссылку на два литературных источника, в которых предлагается характеристика прогрессивных решений уравнения, но с использованием дополнительного аппарата.

3.4. Решение системы параллельных автоматных уравнений

Рассмотрим следующую задачу. Предположим, что существует агент, который вынужден работать в различных контекстах, причем при работе в каждом контексте требуется обеспечить свой уровень сервиса. Если предположить, что контекст, сервис и поведение агента можно описать соответствующими конечными автоматами, то, решая задачу синтеза такого агента, мы придем к системе автоматных уравнений. В мультиагентных системах взаимодействие между компонен-

тами обычно описывается с помощью оператора параллельной композиции, т.е. мы получим систему параллельных автоматных уравнений.

Пусть даны полностью определенные наблюдаемые автоматы S_j и $A_j, j = 1, \dots, n$. Системой параллельных автоматных уравнений называется совокупность уравнений вида:

$$\begin{cases} A_1 \diamond X \equiv S_1 \\ A_2 \diamond X \equiv S_2 \\ \dots \\ A_n \diamond X \equiv S_n \end{cases} \quad (3.2)$$

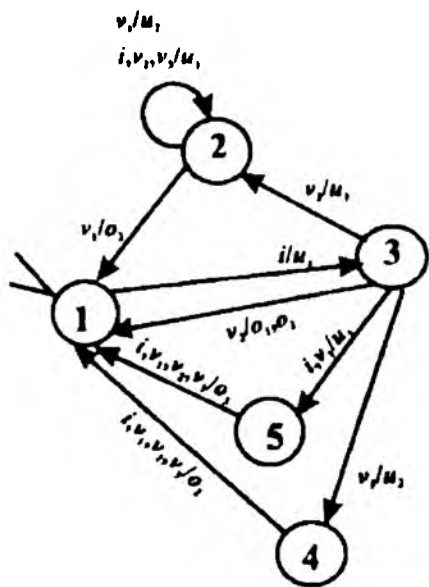
Неизвестная компонента X является автоматом с заданными входным и выходным алфавитами. Автомат B , определенный в алфавитах неизвестной компоненты X , называется *решением системы уравнений* (3.2), если он является решением каждого уравнения системы.

Автомат *Largest* называется *наибольшим решением системы уравнений* (3.2), если любое решение системы уравнений является редукцией автомата *Largest*.

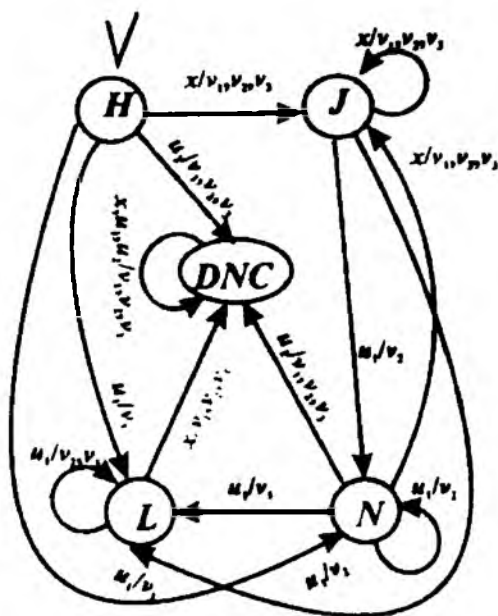
Без ограничения общности мы полагаем, что каждое уравнение соответствует композиции на рис. 1.10.¹⁹ В этом случае компонента X имеет входной алфавит $I_2 \cup U$ и выходной алфавит $O_2 \cup V$. Для решения системы параллельных автоматных уравнений можно использовать алгоритм 2.6, только наибольшее решение каждого уравнения будет находиться для параллельного автоматного уравнения.

Пример. Рассмотрим систему (3.2), состоящую из двух параллельных автоматных уравнений. Пусть автомат S_1 имеет входные символы i и x , выходные символы o_1, o_2, o_3 , одно состояние 1 и переходы $(1, x, o_3, 1), (1, i, o_1, 1), (1, x, o_2, 1)$. Автомат A_1 (рис. 3.7а) имеет внешний входной символ i , внешние выходные символы o_1, o_2, o_3 , внутренние входные символы v_1, v_2, v_3 и внутренние выходные символы u_1, u_2 . Неизвестная компонента имеет внешний входной символ x , внутренние входные символы u_1, u_2 и внутренние выходные символы v_1, v_2, v_3 , т.е. неизвестная компонента не имеет внешних выходов. Наибольшее полностью определенное решение уравнения $A_1 \diamond X \equiv S_1$ показано на рис. 3.7б.

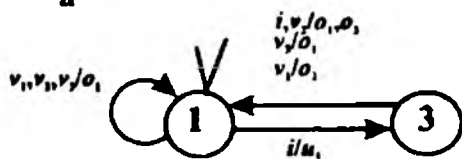
¹⁹ Так же как и в случае системы синхронных автоматных уравнений, топологии могут быть различными для различных S_j и A_j .



а



б



в

Рис.3.7, а автомат A_1

б наибольшее полностью определенное решение уравнения $A_1 \circ X \equiv S_1$

в автомат A_2

Рассмотрим автоматы A_2 и S_2 , где S_2 есть автомат с одним состоянием и переходами $(1, x, o_3, 1)$, $(1, i, o_1, 1)$, $(1, x, o_3, 1)$; автомат A_2 показан на рис. 3.7в. Решим систему из двух уравнений $A_1 \circ X \equiv S_1$ и $A_2 \circ X \equiv S_2$. Непосредственной проверкой можно убедиться, что автомат на рис. 3.7б является наибольшим полностью определенным решением системы уравнений

$$\begin{cases} A_1 \circ X \equiv S_1 \\ A_2 \circ X \equiv S_2 \end{cases}$$

Так же как для уравнений, для систем параллельных автоматных уравнений можно рассматривать частные решения. В списке дополни-

тельной литературы приведены два литературных источника, в которых можно найти более подробное описание, как прогрессивных решений параллельного автоматного уравнения, так и описание других частных решений параллельных автоматных уравнений.

Задачи для самостоятельного решения

1. Разработать алгоритм нахождения наибольшего l -ограниченного решения для системы параллельных автоматных уравнений.

2. Разработать алгоритм нахождения наибольшего полностью определенного муровского решения для системы параллельных автоматных уравнений.

3.5. Контрольные вопросы к главе 3

1. Что такое параллельное автоматное уравнение?

2. Какой автомат называется решением параллельного автоматного уравнения?

3. Какое решение называется наибольшим решением параллельного автоматного уравнения?

4. Какое решение параллельного автоматного уравнения называется полностью определенным решением? l -ограниченным решением? Прогрессивным решением?

5. Как используются полуавтоматы при решении параллельного автоматного уравнения?

6. Как можно найти наибольшее решение параллельного автоматного уравнения в заданных алфавитах с использованием полуавтоматов?

7. В чем основное отличие алгоритмов нахождения наибольшего решения синхронного и параллельного автоматных уравнений?

8. Как найти наибольшее полностью определенное решение параллельного автоматного уравнения? Каким свойством обладает такое решение?

9. Является ли любая полностью определенная редукция наибольшего решения параллельного автоматного уравнения решением уравнения?

10. Как найти наибольшее l -ограниченное решение параллельного автоматного уравнения? Какими свойствами обладает такое решение?

11. При каких условиях любая полностью определенная редукция наибольшего l -ограниченного решения параллельного автоматного уравнения также является решением автоматного уравнения?

12. Является ли любая полностью определенная редукция наибольшего прогрессивного решения параллельного автоматного уравнения решением уравнения? Почему?

4. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ АВТОМАТНЫХ УРАВНЕНИЙ В ЛОГИЧЕСКОМ СИНТЕЗЕ

4.1 Оптимизация автоматной сети на основе решения уравнений

В этом разделе мы приводим примеры оптимизации автоматной сети на основе решения уравнений.

Пример. Рассмотрим композицию автоматов *A*, *B* и *C* на рис. 4.1а. Диаграммы переходов автоматов *A*, *B* и *C* приведены на рис. 4.1б, 4.1в и 4.1г соответственно.

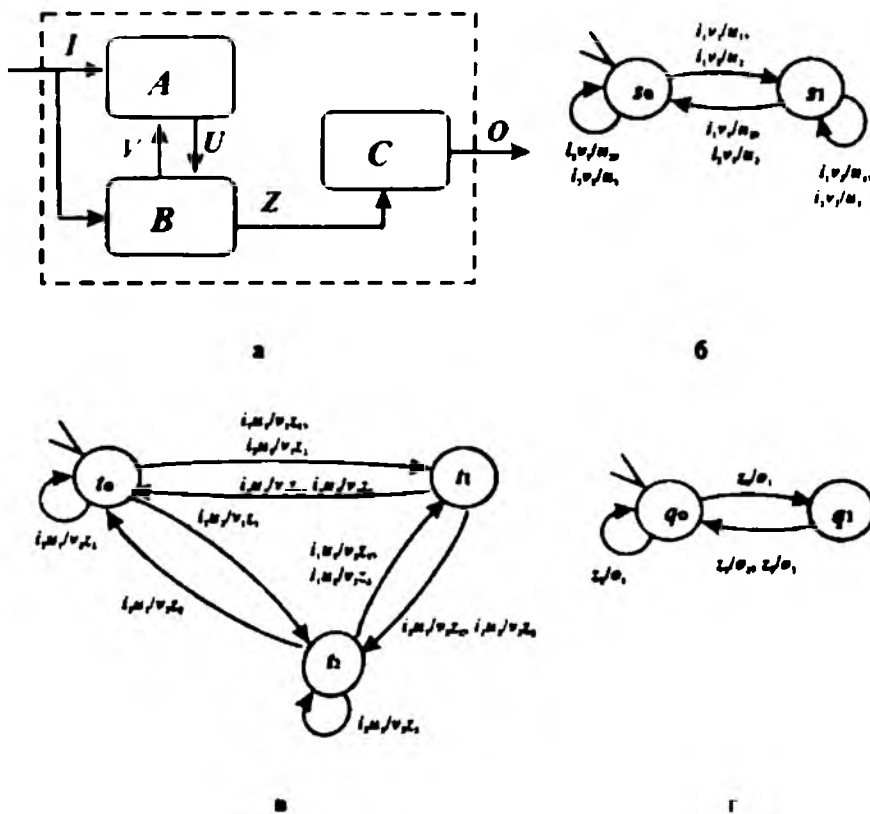


Рис. 4.1, а сеть из трех автоматов
 б диаграмма переходов автомата *A*
 в диаграмма переходов автомата *B*
 г диаграмма переходов автомата *C*

Автомат, описывающий поведение сети на рис. 4.1а, строится как синхронная композиция компонент *A*, *B* и *C*. В этом примере мы не используем полуавтоматное представление и строим композицию

переход за переходом, начиная с начального состояния. За начальное состояние $s_0t_0q_0$ автомата композиции принимается тройка начальных состояний ее компонент. В автомате $A \bullet B \bullet C$ из состояния $s_0t_0q_0$ под действием входного символа i_1 есть переход в состояние $s_1t_1q_1$ с выдачей выходного символа o_1 , если и только если существуют внутренние символы u_1, v_1 и z_1 такие, что $(i_1v_1, s_0, s_1, u_1) \in T_A$, $(i_1u_1, t_0, t_1, v_1z_1) \in T_B$ и $(z_1, q_0, q_1, o_1) \in T_C$. Такие переходы существуют, т.е. четверка $(i_1, s_0t_0q_0, s_1t_1q_1, o_1)$ является переходом композиции. Остальные переходы строятся по такому же правилу. Для данного примера синхронная композиция $A \bullet B \bullet C$ имеет 12 состояний, некоторые из которых являются эквивалентными. Соответствующий приведенный автомат имеет 2 состояния (рис. 4.2).

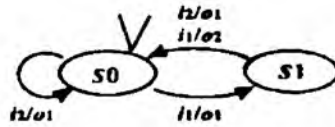


Рис. 4.2. Композиция $A \bullet B \bullet C$

В данном примере, несмотря на наличие обратной связи (рис. 4.1а), композиция является полностью определенным детерминированным автоматом.

Для оптимизации компоненты B нужно решить автоматное уравнение $X \bullet (A \bullet C) = A \bullet B \bullet C$. Наибольшее решение автоматного уравнения имеет три состояния (рис. 4.3а).

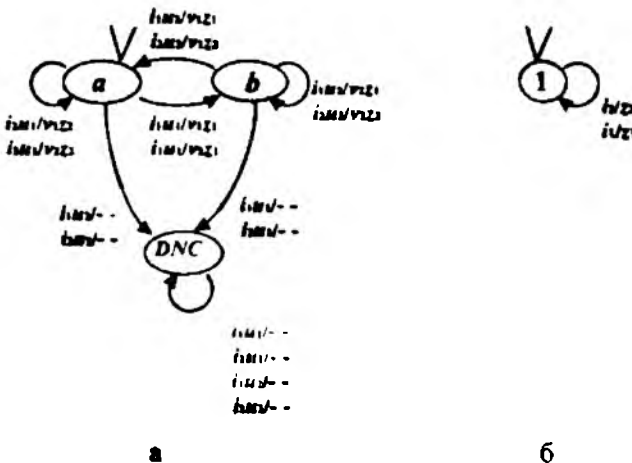


Рис. 4.3, а наибольшее решение уравнения $X \bullet (A \bullet C) = A \bullet B \bullet C$
 б редукция B наибольшего решения

Заметим, что наибольшее решение имеет редукцию B' , которая не зависит от входной переменной u (рис. 4.3б). Таким образом, у компоненты A можно удалить выход, ассоциированный с входной переменной u , в результате чего компонента A становится лишней, так как у нее нет ни одной выходной переменной, значения которой влияют на внешнее поведение сети. Внешний вход I композиции не удаляется, так как он ассоциирован не только с компонентой A , но и с компонентой B . В компоненте B удаляется выход, ассоциированный с входной переменной v компоненты A . Непосредственной проверкой можно убедиться, что композиция $B' \circ C$ эквивалентна композиции $A \circ B \circ C$ (рис. 4.2). Таким образом, в данном примере нам удалось не только оптимизировать компоненты, но и уменьшить количество компонент. Полученная в результате оптимизации автоматная сеть представлена на рис. 4.4.

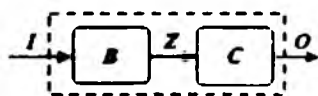


Рис. 4.4. Результат оптимизации автоматной сети на рис. 4.1а

Следует, однако, не забывать, что рассмотренный выше пример является скорее иллюстрацией возможностей оптимизации на основе решения автоматных уравнений. В следующем разделе мы обсуждаем подход к оптимизации, который можно использовать для реальных логических схем.

4.2. Оптимизация комбинационной схемы на основе решения уравнений

В данном разделе мы иллюстрируем, каким образом решение автоматных уравнений может быть использовано для оптимизации логических схем при локальном подходе. Для простоты мы рассматриваем только автоматы без памяти, т.е. комбинационные схемы. Для автоматов с памятью такая оптимизация также возможна, но требует более сложных алгоритмов и больших вычислительных затрат. Таким образом, в данном разделе мы полагаем, что из большой логической схемы вырезан оптимизируемый фрагмент, который различными способами представляется в виде последовательной композиции двух комбинационных схем (рис. 4.5). В каждом фрагменте компоненты оптимизируются на основе решения соответствующего автоматного уравнения. Полученное после оптимизации решение запоминается, если оно

«лучше», чем известное на данный момент решение. Процесс заканчивается, если истекло время на оптимизацию вырезанного фрагмента, или полученное на некотором шаге решение устраивает пользователя.

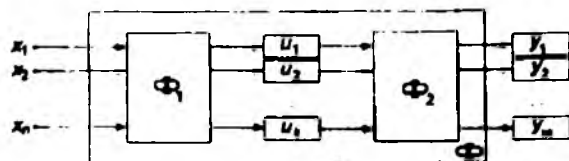


Рис. 4.5. Последовательная композиция двух комбинационных схем

Проиллюстрируем, каким образом автоматное уравнение используется при оптимизации, например, головной компоненты (рис. 4.5). Поскольку выбор оптимального решения из наибольшего решения уравнения является достаточно сложной задачей, мы предлагаем на основе наибольшего решения находить решения (реализации компоненты) с определенными свойствами. В этом разделе мы представляем комбинационные схемы посредством характеристических функций; такое представление на практике часто оказывается более компактным.

Рассмотрим комбинационную схему на рис. 4.5. Каждая из подсхем реализует систему булевых функций (СБФ), и, согласно результатам предыдущих разделов, достаточно часто в качестве каждой из компонент можно использовать схему, реализующую другую СБФ из некоторого множества. Множество допустимых СБФ для головной (или, соответственно, для хвостовой) компоненты композиции описывается как наибольшее решение соответствующего автоматного уравнения (глава 2). В данном разделе мы оптимизируем головную подсхему на основе упрощения функций в соответствующей системе. С этой целью мы выбираем из наибольшего решения, например, такую систему булевых функций (если возможно), в которой одна или несколько функций тождественно равны 0 или 1, или систему, в которой некоторые функции совпадают, или систему, в которой одна из функций есть дизъюнкция (конъюнкция и т.п.) двух других. Процедура выбора оказывается достаточно простой при описании СБФ посредством характеристической функций. В этом случае все описанные ниже проверки можно эффективно выполнить с помощью соответствующего программного обеспечения с использованием двоичных решающих диаграмм (BDD). В списке дополнительной литературы присутствует ссылка на пакет BDD, который имеется в свободном доступе.

Пусть Φ есть система булевых функций (СБФ):

$$\Phi = \begin{cases} u_1 = \varphi_1(x_1, \dots, x_n) \\ \dots \\ u_k = \varphi_k(x_1, \dots, x_n) \end{cases}$$

СБФ Φ можно описать посредством характеристической функции $\Psi_\Phi(x_1, \dots, x_n, u_1, \dots, u_k)$: для набора значений $X_1, \dots, X_m, U_1, \dots, U_k$ переменных $x_1, \dots, x_n, u_1, \dots, u_k$ функция $\Psi_\Phi(X_1, \dots, X_m, U_1, \dots, U_k) = 1$, если и только если $U_1 = \varphi_1(X_1, \dots, X_n), \dots, U_k = \varphi_k(X_1, \dots, X_n)$. Будем говорить, что функция Ψ есть *СБФ-характеристическая* функция, если данная функция является характеристической функцией некоторой СБФ. Характеристическая функция Ψ определяет множество M_Ψ наборов значений переменных, на которых функция равна 1. Если для двух функций θ и Ψ , определенных на одном множестве переменных, справедливо $M_\theta \subseteq M_\Psi$, то будем обозначать этот факт как $\theta < \Psi$.

Рассмотрим композицию из двух комбинационных компонент, реализующих СБФ Φ_1 и Φ_2 (рис. 4.5), и их суперпозицию $\Phi = \Phi_2(\Phi_1)$, которая описывает поведение всей комбинационной схемы. Если СБФ представлены характеристическими функциями $\Psi_{\Phi_1}, \Psi_{\Phi_2}$ и Ψ_Φ , то $\Psi_\Phi = (\Psi_{\Phi_1} \wedge \Psi_{\Phi_2}) \downarrow_{X,Y}$, где $\downarrow_{X,Y}$ обозначает проекцию функции на множество переменных $x_1, \dots, x_m, y_1, \dots, y_m$, т.е. функцию, множество единичных наборов которой есть соответствующая проекция множества единичных наборов функции $(\Psi_{\Phi_1} \wedge \Psi_{\Phi_2})$. Согласно результатам раздела 2, можно найти наибольшее решение уравнения $\Phi = \Phi_2(X)$, которое описывается характеристической функцией $(\Psi_{\Phi_2} \wedge \overline{\Psi_\Phi}) \downarrow_{X,U}$, и сформулировать следующее утверждение.

Теорема 4.1. Для СБФ

$$\Phi_3 = \begin{cases} u_1 = \varphi(x_1, \dots, x_n) \\ \dots \\ u_k = \varphi(x_1, \dots, x_n) \end{cases}$$

справедливо $\Phi_2(\Phi_1) = \Phi_2(\Phi_3)$, если и только если $\Psi_{\Phi_3} < (\Psi_{\Phi_2} \wedge \overline{\Psi_\Phi}) \downarrow_{X,U}$, где $\overline{\Psi}$ обозначает инверсию функции Ψ .

□

Таким образом, теорема 4.1 определяет, какие СБФ могут заменить СБФ Φ_1 на рис. 4.5. Выбирая оптимальную в некотором смысле СБФ Φ_3 , можно упростить головную компоненту исходной комбинационной схемы.

Заметим, что теорему 4.1 можно соответствующим образом переформулировать для случая произвольной композиции без обратных связей, например, для случая, когда хвостовая СБФ зависит не только от промежуточных переменных u_1, \dots, u_k , а также и от (некоторых) входных переменных x_1, \dots, x_n или для случая, когда рассматривается композиция l комбинационных схем, $l > 2$. Соответственно, все описанные в данном разделе результаты применимы и к таким композициям. Мы иллюстрируем, что с использованием характеристических функций выбор оптимальной СБФ решается достаточно просто для ряда оптимизационных критериев.

1. Для последовательной композиции комбинационных компонент, реализующих СБФ Φ_1 и Φ_2 (рис. 4.5), проверить, можно ли выбрать головную СБФ Φ_3 таким образом, что $\Psi_{\Phi_3} \leq (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, и одна из функций системы Φ_3 тождественно равна 1. В этом случае можно синтезировать головную компоненту, которая реализует систему из $(k-1)$ функции.

Согласно теореме 4.1, для решения поставленной выше задачи требуется определить по функции $(\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, существует ли система Φ_3 булевых функций $\theta_1(x_1, \dots, x_n), \dots, \theta_k(x_1, \dots, x_n)$, в которой функция θ_j тождественно равна 1, и $\Psi_{\Phi_3} \leq (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$. Для нахождения системы Φ_3 можно воспользоваться следующим утверждением.

Теорема 4.2. Существует СБФ Φ_3 , $\Psi_{\Phi_3} \leq (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, в которой функция θ_j тождественно равна 1, если и только если функция $((\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u} \wedge u_j)_{\downarrow x}$ тождественно равна единице.

Действительно, конъюнкция $(\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u} \wedge u_j$ равна 1 только на тех единичных наборах функции $(\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, для которых значение переменной u_j равно 1. Соответственно, для того чтобы можно было выбрать СБФ Φ_3 , $\Psi_{\Phi_3} \leq (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, в которой функция θ_j тождественно равна 1, для каждого набора X_1, X_2, \dots, X_n значений переменных x_1, x_2, \dots, x_n должен существовать набор значений U_1, U_2, \dots, U_k переменных u_1, u_2, \dots, u_k , такой, что функция

$(\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u} \wedge u_j$, равна 1 на наборе $x_1, x_2, \dots, x_m, u_1, u_2, \dots, u_k$.

Таким образом, существует СБФ $\Phi_3, \Psi_{\Phi_3} \prec (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, в которой функция θ , тождественно равна 1, если и только если функция $((\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u} \wedge u_j)_{\downarrow x}$ тождественно равна 1.

□

Заметим, что подобным образом можно осуществить проверку возможности выбора СБФ Φ_3 , в которой одна из функций тождественно равна 0. В этом случае достаточно рассмотреть конъюнкцию функции $(\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$ с u_j . Перебирая j от 1 до k , можно проверить, существует ли СБФ Φ_3 , в которой одна или несколько функций тождественно равны 1 или 0. Соответственно, представляет интерес вопрос о нахождении СБФ Φ_3 , которая содержит максимальное число таких функций.

2. Для последовательной композиции комбинационных компонент, реализующих СБФ Φ_1 и Φ_2 (рис. 4.5), проверить, можно ли выбрать головную СБФ Φ_3 с функциями $\theta_1, \dots, \theta_k$ таким образом, что $\Phi_2(\Phi_3) = \Phi_2(\Phi_1)$ и две (или более) функций системы совпадают. В этом случае также можно синтезировать головную компоненту, которая реализует СБФ не из k , а из меньшего числа функций. Для такой проверки в характеристическую функцию можно добавить новую переменную $u_{k+1} = u_i \oplus u_j$. Тогда существует СБФ $\Phi_3, \Psi_{\Phi_3} \prec (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u}$, в которой функции θ и θ_j совпадают, если и только если существует СБФ Φ_4 , зависящая от переменных $x_1, x_2, \dots, x_m, u_1, u_2, \dots, u_k, u_{k+1}$, в которой функция θ_{k+1} тождественно равна 0 и $\Psi_{\Phi_4} \prec (\Psi_{\Phi_1} \wedge \overline{\Psi_{\Phi_2}})_{\downarrow x, u} \wedge (u_{k+1} = u_i \oplus u_j)$. Таким образом, как следствие теоремы 4.2, можно доказать следующее утверждение.

Следствие. Существует СБФ Φ_3 , $\Psi_{\Phi_3} \leftarrow (\Psi_{\Phi_2} \wedge \overline{\Psi_{\Phi_1}})_{\downarrow x, u}$, в которой функции θ_i и θ_j совпадают, если и только если существует СБФ Φ_4 , в которой функция θ_{k+1} тождественно равна θ , и $\Psi_{\Phi_4} \leftarrow (\Psi_{\Phi_2} \wedge \overline{\Psi_{\Phi_1}})_{\downarrow x, u} \wedge (u_{k+1} = u_i \oplus u_j)$.

□

Заметим, что подобным образом можно осуществить проверку возможности выбора СБФ Φ_3 , в которой две (или более) функций системы совпадают с точностью до инверсии. В этом случае нужно проверить, существует ли СБФ Φ_4 , в которой функция θ_{k+1} тождественно равна 1. Перебирая i, j от 1 до k , можно проверить, существует ли СБФ Φ_3 , в которой две или несколько функций совпадают с точностью до инверсии. Соответственно, представляет интерес вопрос о нахождении СБФ Φ_3 с максимальным числом таких функций.

Заметим, что можно сформулировать соответствующие следствия из теоремы 4.2 для проверки возможности выбора СБФ Φ_3 , в которой одна из функций является дизъюнкцией (или конъюнкцией двух других или простой функцией входных переменных) и т.п. В этом случае такую функцию можно реализовать одним простым элементом (конъюнктом, дизъюнктом и т.п.). Таким образом, сократится путь от входов подсхемы к соответствующему выходу, и, следовательно, уменьшится длина пути к некоторым выходам всей схемы. Кроме того, можно будет удалить из подсхемы те элементы, которые влияют на значение только данного выхода, т.е. вполне возможно, что удастся сократить и общее число элементов во всей схеме. Проверки, на основании которых можно выбрать СБФ Φ_3 с необходимыми свойствами для оптимизации головной компоненты, даже для больших схем достаточно быстро выполняются с использованием так называемых SAT-солверов или BDD пакета прикладных программ (см. ссылки в списке дополнительной литературы).

Пример. Проиллюстрируем на простом примере возможности оптимизации на основе теоремы 4.2. Рассмотрим комбинационную схему, представленную на рис. 4.6.

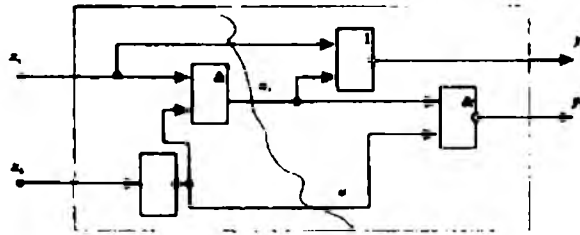


Рис. 4.6. Комбинационная схема, разрезанная на две части

Функцию $(\Psi_{\Phi_2} \wedge \overline{\Psi_{\Phi_1}})_{\downarrow x, U}$ можно представить в виде следующей дизъюнктивной нормальной формы: $(\Psi_{\Phi_2} \wedge \overline{\Psi_{\Phi_1}})_{\downarrow x, U} = \overline{x_1 u_1} \vee \overline{x_2 u_1} \vee \overline{x_1 x_2 u_1 u_2} \vee \overline{x_1 x_2 u_1 u_2}$. Непосредственной проверкой нетрудно убедиться, что конъюнкция этой функции с u_2 есть функция $\overline{x_1 u_1 u_2} \vee \overline{x_2 u_1 u_2} \vee \overline{x_1 x_2 u_1 u_2}$. Таким образом, для наборов 11, 10 и 01 значений переменных x_1, x_2 можно выбрать значение U_1, U_2 из интервала 0-, и, в частности, можно выбрать значение 01. Для набора 00 значений переменных x_1, x_2 можно выбрать значение $U_1 U_2 = 11$, и, таким образом, существует СБФ $\Phi_3, \Psi_{\Phi_1} < (\Psi_{\Phi_2} \wedge \overline{\Psi_{\Phi_1}})_{\downarrow x, U}$, в которой функция $u_2 = \theta_2(x_1, x_2)$ тождественно равна 1. С учетом того, что $u_1 \wedge 1 = u_1$, элемент NAND в исходной схеме можно заменить инвертором, удалив ставшую избыточной одну из связей. Результат оптимизации представлен на рис. 4.7.

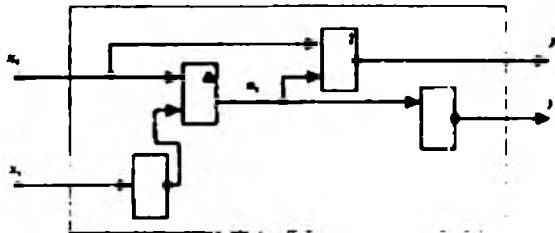


Рис. 4.7. Результат оптимизации схемы на рис. 4.6

Оптимизацию хвостовой компоненты проще осуществить на основе безразличных последовательностей. С этой целью можно сначала выбрать для головной компоненты СБФ, которая имеет достаточно мало различных выходных значений. На следующем этапе находится характеристическая функция всей схемы, зависящая как от внешних входных и внешних выходных переменных, так и входных перемен-

ных хвостовой подсхемы. Проекция этой функции на входные и выходные переменные хвостовой подсхемы опишет ту часть поведения хвостовой подсхемы, которая влияет на поведение всей схемы и не может быть изменена. На всех остальных входных наборах поведение хвостовой компоненты может быть определено произвольно.

Теперь можно проверить, нельзя ли с помощью подходящего доопределения поведения на безразличных наборах выбрать для хвостовой компоненты СБФ, в которой одна из функций является дизъюнкцией (или конъюнкцией двух других или входных переменных) и т.п. В этом случае такую функцию можно реализовать одним простым элементом (конъюнктом, дизъюнктом и т.п.). Последствия для оптимизации будут такие же, как и для головной компоненты: сократится путь от входов подсхемы к соответствующему выходу, и, следовательно, уменьшится длина пути к некоторым выходам всей схемы. Кроме того, можно будет удалить из подсхемы те элементы, которые влияют на значение только данного выхода, т.е. вполне возможно, что удастся сократить и общее число элементов во всей схеме.

В заключение заметим, что результаты раздела 4.2 можно распространить на последовательностные схемы. В этом случае поведение каждой из компонент композиции описывается некоторым структурным автоматом, и помимо функций выходов каждой компоненты необходимо рассматривать еще и функции переходов (функции возбуждения). Множество допустимых структурных автоматов для компоненты (наибольшее решение соответствующего автоматного уравнения) может быть найдено по характеристическим функциям так же как и для комбинационных схем. Однако в случае последовательностных схем в наибольшем решении могут появиться состояния, в которых поведение определено не для каждого входного набора, и такие состояния (и переходы в них) необходимо итеративно удалить до начала оптимизации. Для проверки возможности представления одной из выходных функций суперпозицией некоторых простых функций также необходимо проверить, что такая суперпозиция существует в каждом из оставшихся достижимых состояний.

4.3. Пример протокольного конвертора на основе решения уравнений

В качестве иллюстрации изложенного выше метода решения параллельного автоматного уравнения рассмотрим синтез конвертора

для двух несовместимых протоколов²⁰. В качестве примера мы рассматриваем несовместимые alternating-bit (AB) и non-sequenced (NS) протоколы. Система связи состоит из передатчика, использующего протокол AB, приемника, использующего протокол NS, и должна передавать данные по каналу, который работает в обоих направлениях, но по очереди.

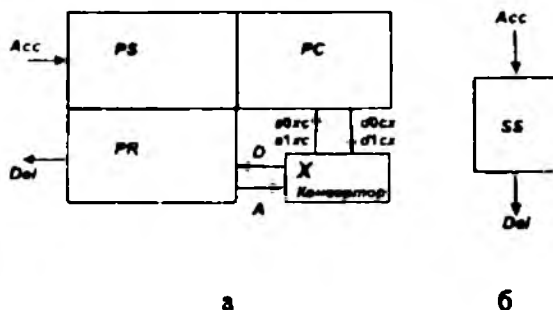


Рис. 4.8, а блок-диаграмма всей системы
 б описание системы с точки зрения пользователя

Рис. 4.8а показывает блок-схему всей системы. Каждая компонента представлена прямоугольником с входящими и выходящими стрелками, чтобы показать, соответственно, входные и выходные символы. Передатчик состоит из AB протокола (*PS*) и его канала передачи (*PC*), а приемник *PR* использует протокол NS. Конвертор должен связать все части *PS*, *PC* и *PR* так, чтобы полученная система удовлетворяла спецификации (*SS*). События *Acc* и *Del* представляют интерфейс системы с пользователем. Конвертор *X* переводит сообщение передатчика (с использованием AB протокола) в формат, который понимает приемник (NS протокол). Например, подтверждение сообщений *A*, полученных конвертором от приемника, трансформируются в подтверждения alternating-bit протокола (*a0xc* для подтверждения нулевого бита и *a1xc* для подтверждения единичного бита) и проходят по каналу к передатчику (*a0xc* для подтверждения нулевого бита и *a1xc* для подтверждения единичного бита); данные проходят от передатчика в канал (*d0sc* для сообщений, контролируемых нулевым битом и *d1sc* для сообщений, контролируемых единичным битом), а из канала к конвертору (*d0cx* для сообщений, контролируемых нулевым битом, и *d1cx* для

²⁰ Пример взят из готовящейся к публикации монографии T Villa, N.Yevtushenko, R.K.Brayton, A.Mishenko, A.Petrenko, A.Sangiovanni-Vincentelli « The Problem of the Unkown Component from Theory to Applications».

сообщений, контролируемых единичным битом), чтобы быть преобразованными конвертором в сообщение с данными для приемника.

Мы сразу представляем поведение компонент полуавтоматами вида P^j , у которых каждый переход исходного автомата «растянут» на два перехода: первый помечен входным сигналом (символ ?), второй – выходным сигналом (символ !). Эти полуавтоматы показаны на рис. 4.9.

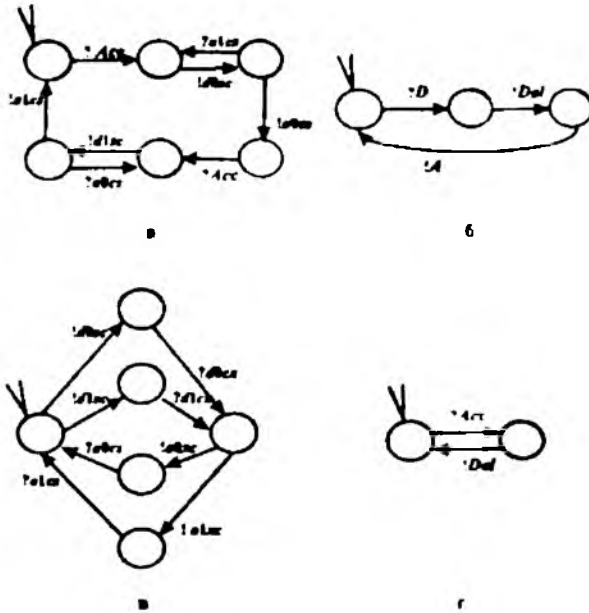


Рис. 4.9, а диаграмма переходов передатчика PS
 б диаграмма переходов приемника PR
 в диаграмма переходов канала передачи PC
 г полуавтомат, описывающий поведение всей системы

Рис. 4.10 показывает наибольшее префикс замкнутое решение $PS \diamond PC \diamond PR \diamond L_{SS}^{\sim}$ для конвертора.

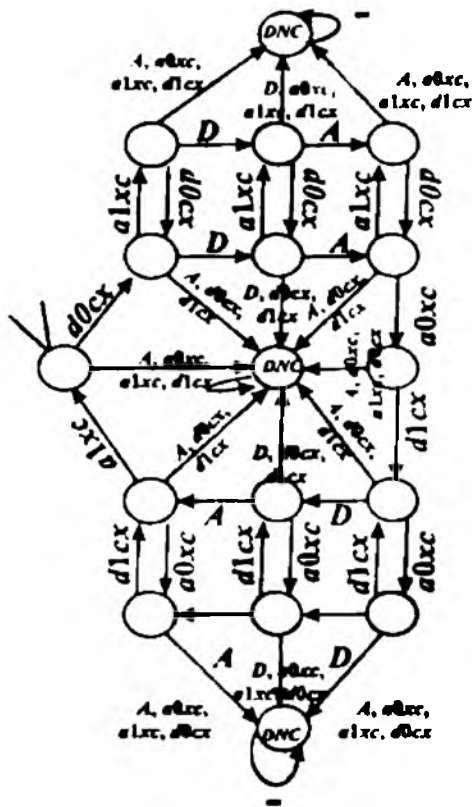


Рис. 4.10. Наибольшее префикс-замкнутое решение для конвертора

На рис. 4.11 мы приводим наибольшее 2-ограниченное решение, как пример решения с ограничением на число внутренних коммуникаций.

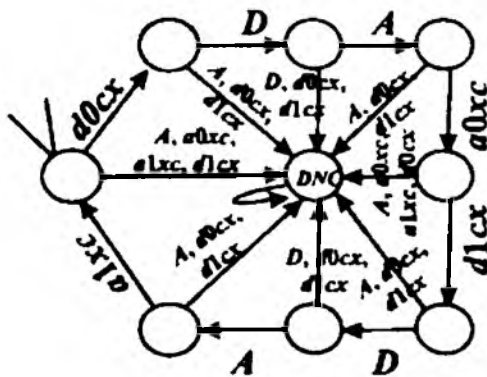


Рис. 4.11. Наибольшее 2-ограниченное решение

Рис. 4.12 показывает композицию конвертора и остальных частей, которая демонстрирует, что наибольшее решение хотя и является про-

грессивным, однако не всякая его полностью определенная редукция обладает таким свойством. Такое возможно, если, например, после входного действия $d0cx$ конвертор всегда выбирает $a1xc$ (цикл $d0cx$ - $a1xc$), т.е. появится тупик после внешнего входного сигнала Acc . Однако после $d0cx$ из цикла есть другой выход D (кроме $a1xc$), так что наибольшее решение, тем не менее, прогрессивное. Как уже отмечалось ранее, цикл удалить нельзя, иначе мы потеряем решение с последовательностями $(d0cx a1xc)^k D$, где k – натуральное число.

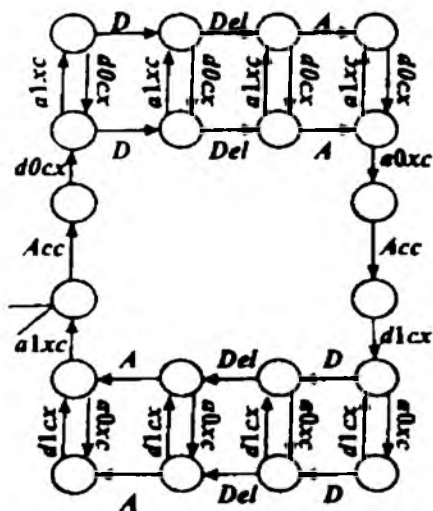


Рис. 4.12. Композиция $PS \diamond PC \diamond PR$ с наибольшим решением уравнения

ЛИТЕРАТУРА

1. Евтушенко Н.В., Петренко А.Ф., Ветрова М.В. Недетерминированные автоматы: анализ и синтез. Ч. 1.
2. Отношения и операции Учебное пособие. – Томск: Томский государственный университет, 2006. – 142 с.
3. Евтушенко Н., Вилла Т., Петренко А., Брайтон Р., Санджованни-Винцентелли А. Решение уравнений в логическом синтезе. – Препринт. Томск: Изд-во «Спектр», Томск, 1999. – 27 с.
4. Kam T., Villa T., Brayton R., Sangiovanni-Vincentelli A. Synthesis of finite state machines: functional optimization. – Kluwer Academic Publishers, 1997. – 296 p.
5. Yevtushenko N., Villa T., Brayton R., Petrenko A. and Sangiovanni-Vincentelli A. Solution of synchronous language equations for logic synthesis // Вестник Томского государственного университета. Приложение. – 2002. – № 1 (II). – С. 132–138.

Дополнительная литература

1. Бушков В.Г. К описанию прогрессивных решений параллельного автоматного уравнения // Прикладная дискретная математика. – Томск, 2008. – №1(1). – С. 120–125.
2. Буффалов С.А. Исследование живых и безопасных решений параллельных уравнений и неравенств на множестве полуавтоматов и автоматов: Дис. ... канд. техн. наук. – Томск, 2002. – 145 с.
3. Ветрова М.В. Разработка алгоритмов синтеза и тестирования конечно автоматных компенсаторов: Дис. ... канд. техн. наук. – Томск, 2004. – 160 с.
4. Вилла Т., Евтушенко Н., Жарикова С. Характеризация живых решений синхронного автоматного уравнения // Вестник Томского государственного университета. – Сентябрь 2003. – № 278. – С. 129–133.
5. Жарикова С.В., Евтушенко Н.В. Решение автоматного уравнения для многомодульной композиции // Вестник Томского государственного университета. Приложение. – Томск, 2007. – №23. – С. 62–65.
6. Жарикова С.В., Евтушенко Н.В. Решение автоматных уравнений в различных приложениях // Вестник Красноярского государственного университета. – 2004. – №3. – С. 35–39.

7. Жарикова С., Евтушенко Н., Ветрова М. Оптимизация декомпозиций конечных автоматов на основе решения автоматных уравнений // Материалы XV Международной школы-семинара «Синтез и сложность управляющих систем». – Новосибирск, 2004 – С. 29–34.
8. Жарикова С. Оптимизация элементов цифровых схем посредством решения систем автоматных уравнений // Вестник Томского государственного университета. Приложение. – 2002. – № 1 (II). – С. 255–259.
9. Евтушенко Н.В., Ветрова М.В. Синтез контроллеров для абстрактных и структурных автоматов // Известия АН: Теория и системы управления. – 2004. – №6. – С. 46–51.
10. Кушик Н.Г., Рекун М.В. К оптимизации комбинационных схем на основе решения уравнений // Журнал Сибирского федерального университета «Mathematics & physics». – Красноярск, 2008. – №1(3). – С. 290–295.
11. Тихомирова С.В. Оптимизация многокомпонентных дискретных систем на основе решения автоматных уравнений: Дис. ... канд. техн. наук. – Томск, 2008. – 145 с.
12. Buffalov S., Yevtushenko N. Studying solutions to a parallel FSM equation // Bulletin of the Novosibirsk computing center. – Computer science. – 2001. – V.11.– P. 7–16.
13. CUDD [Electronic resource] – режим доступа: <http://vlsi.colorado.edu/~fabio/CUDD/>
14. El-Fakih Kh., Yevtushenko N. Progressive Solutions to FSM Equations // Proceedings of the CIAA, 2008. – P. 274–282.
15. El-Fakih Kh., Yevtushenko N., Buffalov S., Bochmann G.v. Progressive solutions to a parallel automata equation // Theoretical computer science. – 2006. – № 362(1-3). – P. 17–32.
16. Eliseeva N., Jiang J.-H.R., Kushik N., Yevtushenko N. Symmetrization in digital circuit optimization // Proceedings of IEEE East-West Design and Test Symposium. – Kharkov, 2009. – P. 103–106.
17. Gromov M., Kushik N. Software package for optimizing digital circuits // Proceedings of the third spring young researchers colloquium on software engineering. – Moscow, 2009. – P. 68–70.
18. Kim J. and Newborn M.M. The simplification of sequential machines with input restrictions // IRE transactions on electronic computers. – 1972. – P. 1440-1443.
19. Kushik N., Sapunkov G., Prokopenko S., Yevtushenko N. Minimizing path length in digital circuits based on equation solving // Proceedings

- of IEEE East-West Design & Test Symposium. – Lviv, 2008. – P. 365–370.
20. Merlin P., Bochmann G.v. On the construction of submodule specification and communication protocols // ACM transactions on programming languages and systems. – 1983. –5(1). – P. 1–25.
 21. Petrenko A., Yevtushenko N. Solving asynchronous equations // Formal Description Techniques and Protocol Specifications, Testing and Verification, FORTE XI/PSTVXIII. – 1998. – P. 231–247.
 22. Yevtushenko N., Villa T., Brayton R.K., Petrenko A., Sangiovanni-Vincentelli A. Compositionally progressive solutions of synchronous FSM equations // Discrete event dynamic systems. – 2008. – № 18(1). – P. 51–89.
 23. Yevtushenko N., Villa T., Zharikova S. Solving language equations over synchronous and parallel composition operators // Proceedings of the International workshop on theory and applications of language equations. – Finland, 2007. – P. 14–32.
 24. Yevtushenko N., Zharikova S., Vetrova M. Multi component digital circuit optimization by solving FSM equations // Proceedings of the Euro-micro symposium on digital system design. – Turkey, 2003. – P. 62–68.
 25. Yevtushenko N., Villa T. Brayton R.K., Petrenko A., Sangiovanni-Vincentelli A. Solving a parallel language equation // Proceedings of the ICCAD. – USA, 2001. – P. 103–110.

Отпечатано на участке оперативной полиграфии
редакционно-издательского отдела ТГУ

Заказ № 3 от «13» 04 2010 г. Тираж 100 экз.