ИЗВЕСТИЯ АКАДЕМИИ НАУК. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 1999, № 5, с. 94–100

ПРИНЯТИЕ РЕШЕНИЙ И ЭКСПЕРТНЫЕ СИСТЕМЫ

УДК 658.012.011.56

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ: ПРЕОДОЛЕНИЕ НЕОПРЕДЕЛЕННОСТИ НА ИСПОЛНИТЕЛЬНОМ УРОВНЕ*

© 1999 г. В. Н. Захаров

Москва, ВЦ РАН

Поступила в редакцию 30.05.99

Расссматриваются средства преодоления неопределенности информации при описании автоматных моделей, используемых на исполнительном уровне интеллектуальных систем логического управления. Предложен специальный язык (продукционного типа) описания моделей исполнительного уровня, названный языком логических продукций. Приведено описание этого языка и продемонстрированы его возможности для представления сильно недоопределенных автоматов. Приведены примеры работы алгоритмов минимизации продукционных описаний на этом языке.

Введение. Под интеллектуальными системами логического управления подразумеваются управляющие системы с многоуровневой архитектурой, на нижних уровнях иерархии которых используются формальные логические модели. Управляющие структуры систем такого класса строятся в соответствии с принципом “более высокому уровню в иерархической структуре соответствует более высокая степень интеллектуальности и наоборот” [1]. Поскольку понятие “интеллектуальность” в общепринятом смысле является неформализуемым, обычно под “интеллектуальностью системы”, понимается способность системы работать с базой внешних событий (ситуаций) в условиях неполной определенности поступающей входной информации. А под “степенью интеллектуальности” – степень развитости используемых средств борьбы с неопределенностью на основе знаний закономерностей проблемной среды, или внешнего мира, в котором функционирует данная управляющая система. Таким образом, исполнительный уровень (уровень формальных моделей) относят к уровням с нулевой степенью интеллектуальности, т.е. к уровням, неспособным к работе с неточной и с неопределенной информацией. Обработка неопределенной информации осуществляется на самых верхних уровнях иерархической структуры с использованием таких средств, как индуктивный вывод, правдоподобные рассуждения, вывод по аналогии, на основе здравого смысла, использование средств обучения и т.п.

Наиболее распространенными формальными моделями, используемыми на исполнительных уровнях интеллектуальных систем логического управления, являются модели конечных автоматов (комбинационных схем и схем с памятью). При логическом синтезе исполнительных устройств приходится также сталкиваться с недоопределенностью используемой информации. Однако эта неопределенность несколько другого рода. Если на верхних уровнях иерархической структуры приходится в условиях неопределенной информации принимать управленческие решения и на основе принятых решений формировать управляющие воздействия для исполнительных механизмов нижних уровней, то неопределенность используемой информации при синтезе исполнительных логических устройств не влияет на формирование управляющих воздействий в процессе функционирования системы. Она как бы устраняется при программной или аппаратной реализации моделей устройств управления на этапе их синтеза. Но на процессы самого логического синтеза так же, как на результаты синтеза (сложность и надежность полученных устройств) эта неопределенность конечно сказывается.

В данной статье анализируются затронутые вопросы и вводится специальный язык, удобный для описания моделей сильно недоопределенных автоматов.

1. Ограниченность классических методов описания автоматных моделей. Методология проектирования логических устройств на базе автоматных моделей известна с давних времен и в общих чертах представляет собой следующую совокупность процедур [2]. Исходными данными при логическом синтезе служат описания работы проектируемого устройства, представленные в любом виде, например, в виде диаграммы, или в графической форме или в виде текста на естественном языке (вербальное описание). На первом этапе

* Работа выполнена при финансовой поддержке РФФИ (грант 99-01-00150а).

синтеза исходные данные формализуются (см. рис. 1). Результатом этого процесса служит построение формальной модели проектируемого устройства. В случае использования модели конечного автомата в качестве формальной модели рассматривается автоматная таблица или автоматный граф. Далее выполняется анализ полученной формальной модели, ее минимизация, кодирование и переход к таблицам, задающим функции выходов и внутренних переменных (в случае модели автомата с памятью). Следующий этап состоит в построении компактных аналитических выражений для полученных функций (систем функций), по которым и производится построение логической структуры, реализующей исходное описание проектируемого устройства. Заключительный этап синтеза – построение реальной структуры проектируемого устройства управления, которая отличается от полученной на предыдущем этапе логической структуры учетом реальных ограничений, присущим реальным элементам структуры.

Основной трудностью при реализации всех этих этапов логического синтеза является проклятие размерности. Размерность и недоопределенность автоматных таблиц очень быстро растет с ростом числа входных и внутренних переменных автомата. Например, если требуется построить автоматную таблицу логического устройства с десятью входами и десятью внутренними кодовыми переменными, то такая таблица должна содержать более миллиона клеток. Причем даже, если бы мы попытались начать строить такую таблицу (например, с помощью какого-либо алгоритма в памяти машины) при начальном ее заполнении подавляющее большинство клеток осталось бы пустыми, поскольку среди более, чем миллиона клеток, технологически допустимых комбинаций (пар: входной символ – состояние) оказалось бы не так уж много. А между тем устройство с десятью входами и десятью кодовыми переменными состояний нельзя считать чрезвычайно сложным. На практике встречаются и более сложные случаи.

Этап "Определение автоматных функций" (см. рис. 1) также вызывает затруднения. Как будет показано в третьем разделе, аналитических представлений в виде дизъюнктивных и конъюнктивных нормальных форм, а также произвольных скобочных форм для недоопределенных булевой функции не существует. Получается, что традиционные средства аналитического представления выражений булевой алгебры оказываются непригодными для аналитического представления не полностью определенных булевых функций. А именно с такими функциями проектировщик дискретных устройств как правило и имеет дело. Поэтому и возникает очень важная проблема –

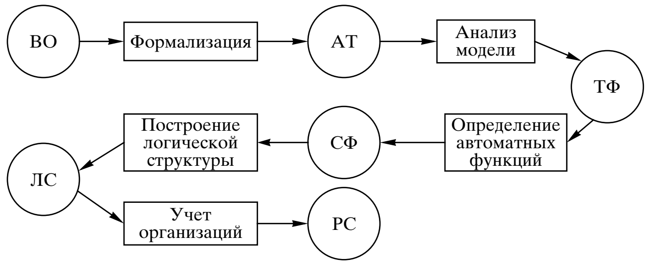

Рис. 1. Этапы проектирования устройств логического управления: ВО – вербальное описание; АТ – автоматная таблица; ТФ – таблица функций; СФ – система функций; ЛС – логическая структура; РС – реальная структура.

разработка новых языковых средств описания сильно недоопределенных автоматов.

И еще одно небольшое замечание. Имея изначально дело с не полностью определенными булевыми функциями, проектировщик в результате своей работы получает программную или аппаратную их реализацию. Полученной структуре можно поставить в соответствие аналитическое выражение реализуемым функциям, которые окажутся полностью определенными. Поэтому закончим обсуждение особенностей реализации не доопределенных условий работы автоматов неким общим утверждением, которое назовем принципом реализации.

П р и н ц и п р е а л и з а ц и и: "Все, что реализовано, то полностью определено".

В соответствии с этим принципом утверждение о том, что построенная схема реализует заданную не полностью определенную функцию или заданный недоопределенный автомат, некорректно. В данном простом случае этот факт в достаточной мере очевиден. Однако в некоторых более сложных случаях это не столь очевидно.

Итак, каждая реализация соответствует полностью определенной в рамках выбранной модели входной информации. Поскольку разработчик логических устройств исполнительного уровня пользуется на начальных стадиях проектирования не полностью определенной информацией, возникает задача разработки языковых средств для представления в аналитической (не табличной) форме не полностью определенных автоматных функций.

2. Язык продукционных правил. Одним из наиболее распространенных языковых средств представления знаний в искусственном интеллекте является язык продукционных правил. В самом общем случае под продукционным правилом, или иначе, под продукцией подразумевается запись вида $\mathbf{A} \longrightarrow \mathbf{B}$, где **A**, **B** – условия или действия. Смысл этой записи состоит в том, что если имеет место условие **A**, то имеет место и условие (или действие) **B**. Если же условие **A** не выполняется, то об условии или действии **B** ничего определенного сказать нельзя. Эта особенность является крайне важной, поскольку позволяет организовать работу базы знаний, содержащей несколько десятков, а иногда и значительно большее число продукционных правил, составляющих продукционную систему. Неопределенность условия или действия **B** объясняется тем фактом, что, с одной стороны, условие **B** не должно иметь места при отсутствии условия **A**, а с другой – условие **B** может иметь место, если в базе правил найдется правило вида **C** $\longrightarrow$ **B** и будет выполняться условие **C**. База знаний эффективно работающей системы должна быть устроена таким образом, чтобы для каждой анализируемой внешней ситуации нашлось бы условие для срабатывания какого-либо правила (полнота базы правил), а кроме того, никакое условие не должно приводить к срабатыванию двух и более правил (непротиворечивость базы правил).

Рассмотрим теперь частный случай продукции – правило вида $\varphi \vdash \psi$, где φ, ψ – булевы функции, а знак $\vdash$ имеет тот же смысл, что и стрелка в продукционном правиле, а его использование лишь подчеркивает особенность продукционного правила, в правой и левой части которого находятся булевы функции. Такой специальный вид продукции назовем логической продукцией (или секвенцией [3]). Смысл логической продукции состоит в том, что если левая часть $\varphi = 1$, то правая часть $\psi = 1$. Если же левая часть $\varphi = 0$, то о левой части $\psi = ?$ ничего определенного сказать нельзя.

3. Продукционное представление булевых функций. Рассмотрим особенности аналитического представления булевых функций в совершенной дизъюнктивной нормальной форме. Следующее утверждение является общеизвестным.

Утверждение 1. Любая полностью определенная булева функция за исключением функции "константа ноль" (принимающая значения ноль на все возможных наборах двоичных переменных) может быть представлена в совершенной дизъюнктивной нормальной форме (СДНФ) следующего вида:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{i \in N_1} F_i, \qquad (3.1)$$

где F_i – характеристические функции единицы, определяемые как

$$F_i = x_1^{\sigma_1} \,\&\, x_2^{\sigma_2} \,\&\, \ldots \,\&\, x_n^{\sigma_n} = \begin{cases} 1, & \text{если номер набора есть } i, \\ 0, & \text{в противном случае}, \end{cases}$$

$x_i^{\sigma_i}$ – переменные со знаком инверсии или без него:

$$x_i^{\sigma_i} = \begin{cases} x_i, & \text{если } \sigma_i = 1; \\ \bar{x}_i, & \text{если } \sigma_i = 0; \end{cases}$$

$\sigma_i \in \{0, 1\}$ – значение i-й переменной во входном наборе; N_1 – множество двоичных номеров входных наборов двоичных переменных, на которых заданная функция принимает единичные значения. Двоичный номер входного набора определяется следующим образом:

$$i = \sigma_1 2^{n-1} + \sigma_2 2^{n-2} + \ldots + \sigma_{n-1} 2^1 + \sigma_n 2^0.$$

Рассмотрим в качестве примера три функции, заданные следующей таблицей (см. таблицу).

Первая функция $f_1(x_1, x_2, x_3)$ является сильно недоопределенной (она определена всего на четырех наборах из восьми). Вторая $f_2(x_1, x_2, x_3)$ и третья $f_3(x_1, x_2, x_3)$ функции – полностью определенные. Совершенная дизъюнктивная нормальная форма (3.1) для второй функции выглядит следующим образом:

$$f_2(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_3. \qquad (3.2)$$

Для третьей функции более компактным представлением будет служить выражение в совершенной конъюнктивной нормальной форме вида:

$$f_3(x_1, x_2, x_3) = (x_1 \vee \bar{x}_2 \vee x_3) \,\&\, (\bar{x}_1 \vee \bar{x}_2 \vee x_3).$$

Если попытаться воспользоваться формулой (3.1) для представления первой функции $f_1(x_1, x_2, x_3)$, то мы получим аналитическое выражение, аналогичное (3.2):

$$f_1(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_3 = f_2(x_1, x_2, x_3),$$

т.е. аналитическое выражение для полностью определенной булевой функции. Отсюда следует очевидное утверждение.

Утверждение 2. Не полностью определенная булева функция собственного аналитического представления в виде совершенных дизъ-

Таблица

i	x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	0	–	0	1
1	0	0	1	1	1	1
2	0	1	0	0	0	0
3	0	1	1	–	0	1
4	1	0	0	–	0	1
5	1	0	1	1	1	1
6	1	1	0	0	0	0
7	1	1	1	–	0	1

юнктивных и конъюнктивных нормальных форм не имеет. Попытка представления недоопределенной булевой функции в любой из указанных форм приводит к доопределению функции нулями в случае использования совершенной дизъюнктивной нормальной формы и единицами в случае использования совершенной конъюнктивной нормальной формы.

Попытаемся получить аналитическое представление недоопределенной булевой функции с помощью логических продукционных правил следующим образом.

Для множества номеров наборов N_1, на которых не полностью определенная булева функция принимает единичные значения, справедливо следующее соотношение:

$$\bigvee_{i \in N_1} F_i \vdash f(x_1, x_2, \ldots, x_n). \tag{3.3}$$

В соответствии с семантикой логической продукции эта запись указывает лишь на то, что на любом наборе из множества N_1 функция $f(x_1, x_2, \ldots, x_n)$ принимает единичные значения. А на наборах множества N_0 функция $f(x_1, x_2, \ldots, x_n)$ не определена.

Для множества номеров наборов N_0, на которых не полностью определенная булева функция принимает нулевые значения, справедливо следующее соотношение:

$$\bigvee_{j \in N_0} F_j \vdash \bar{f}(x_1, x_2, \ldots, x_n). \tag{3.4}$$

Это соотношение указывает на то, что функция $f(x_1, x_2, \ldots, x_n)$ на любом наборе из множества N_0 принимает нулевые значения. Естественно, что для наборов, на которых заданная таблицей функция не определена, соответствующие соотношения отсутствуют.

Соотношения (3.3) и (3.4) в совокупности будем называть продукционным представлением не полностью определенной булевой функции. Заметим, что для полностью определенной булевой функции эти соотношения также пригодны, но являются избыточными. В самом деле, для задания полностью определенной булевой функции вполне достаточно либо выражения (3.3), либо выражения (3.4). В первом случае знак логической продукции заменяется знаком равенства и мы имеем дело с совершенной дизъюнктивной нормальной формой. Во втором случае после замены знака логической продукции на знак равенства и применения к обоим частям равенства правила де Моргана получим совершенную конъюнктивную нормальную форму.

Компактность продукционного представления не полностью определенной булевой функции будет тем большей, чем больше ее неопределенность. Кроме того, можно попытаться минимизировать аналитические выражения в левых частях секвенций, используя для этой цели известные правила минимизации функций в классе дизъюнктивных нормальных форм (склеивания и поглощения).

В качестве примера приведем минимальное продукционное аналитическое выражение для функции $f_1(x_1, x_2, x_3)$, заданной таблицей:

$$\bar{x}_2 x_3 \vdash f_1(x_1, x_2, x_3),$$

$$x_2 \bar{x}_3 \vdash \bar{f}_1(x_1, x_2, x_3).$$

Эти соотношения получены путем склеивания конъюнкций в левых частях каждой логической продукции.

4. Продукционное представление автоматов с памятью. Одним из наиболее распространенных способов описания работы автомата с памятью служит представление его работы с помощью автоматной таблицы. Автоматная таблица, как известно, представляет собой матрицу, столбцам которой ставятся в соответствие входные символы автомата, а строчкам – его внутренние состояния. Клетки автоматной таблицы на пересечении столбцов и строчек заполняются новыми состояниями, в которые переходит автомат под воздействием изменившихся значений входов, и новыми значениями выходов. Таким образом, автоматная таблица задает две автоматные функции: функцию переходов δ и функцию выходов λ, определяемые соответственно:

$$\delta\colon \mathbf{X} \times \mathbf{Q} \longrightarrow \mathbf{Q}; \quad \lambda\colon \mathbf{X} \times \mathbf{Q} \longrightarrow \mathbf{Y},$$

где $\mathbf{X}$, $\mathbf{Q}$, $\mathbf{Y}$ – множества входных символов, внутренних состояний и выходных символов автомата соответственно.

Рассмотрим автомат с n входами, m выходами и r кодовыми переменными состояний. Тогда для описания его функций переходов и выходов, повидимому потребуются логические продукции следующих двух видов:

$$\begin{aligned} &(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)_i (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_r)_j \vdash (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_r)_k, \\ &(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)_i (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_r)_k \vdash (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)_l, \end{aligned} \tag{4.1}$$

где $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)_i$, $(\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_r)_k$ и $(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)_l$ – i-й входной символ, k-е внутреннее состояние и l-й выходной символ автомата соответственно. Логическая продукция первого типа служит для описания функции переходов автомата, а второго типа – для представления функции его выходов. Представление автоматной таблицы системой логических продукций (4.1) будем называть продукционным описанием автомата с памятью.

Конечно замена автоматной таблицы на систему продукций с числом записей, равным числу клеток автоматной таблицы никого в восторг не приведет. Однако, если автоматная таблица весьма громоздка в силу большого числа входов и кодовых переменных состояний, а также особенно в случае высокой степени неопределенности автомата (большинство клеток таблицы – пустые) рассматриваемое продукционное представление автоматных функций может оказаться более предпочтительным, чем таблица. К тому же для продукционных представлений можно предложить эффективные способы минимизации, основанные на выполнении следующих двух операций упрощения [3]: операции замены двух и более логических продукций с одинаковыми правыми частями на одну продукцию (операция 1) и операции исключения из правой части продукции общего для правой и левой частей конъюнктивного множителя (операция 2). Приведем эти операции:

Операция 1.

$$\begin{matrix}\varphi_1 \vdash \psi \\ \varphi_2 \vdash \psi\end{matrix} \Leftrightarrow \varphi_1 \vee \varphi_2 \vdash \psi.$$

Эта операция позволяет получить продукцию, описывающую в общем случае несколько (в данном случае два) переходов. В левой части полученной обобщенной продукции полезно попытаться произвести возможные упрощения.

Операция 2.

$$\varphi_1\varphi_2 \vdash \varphi_2\psi \Leftrightarrow \varphi_1\varphi_2 \vdash \psi.$$

Операция пригодна для сокращения продукционного описания функций переходов автомата, так как только в продукциях такого вида могут иметь место общие конъюнктивные множители (кодовые переменные состояния).

Заметим, что в соответствии с семантикой логических продукционных правил для продукций справедлива и операция вида

$$\begin{matrix}\varphi \vdash \psi_1 \\ \varphi \vdash \psi_2\end{matrix} \Leftrightarrow \varphi \vdash \psi_1 \vee \psi_2.$$

Однако, для описания автоматов эта операция не допустима. Смысл получаемой в результате этой операции обобщенной продукции состоит в том, что автомат под воздействием одного и того же сигнала из одного состояния может переходить в разные состояния, что для автомата недопустимо. Поэтому продукции с дизъюнкцией конъюнкций в правой части называются "неавтоматными". Автомат существенно последовательная модель. Неавтоматные продукции не бесполезны. Они пригодны, например, для описания параллельных процессов.

5. Минимизация продукционных описаний автоматов с памятью. В структурном отношении автомат с памятью может быть представлен совокупностью двух взаимосвязанных блоков: блоком, реализующим функции переходов от состояния к состоянию и логическим преобразователем, представляющим собой обычную комбинационную схему, реализующую функции выходов. Каждый из блоков может быть описан с применением соответствующих логических продукций. Эти продукционные описания можно минимизировать путем применения рассмотренных выше двух операций над логическими продукциями. При этом минимизация описания первого блока осуществляется с использованием двух операций, а для минимизации описания блока формирования функций выходов пригодна только первая операция, поскольку в продукциях, описывающих комбинационную схему, не могут встречаться общие конъюнктивные множители для левых и правых частей.

Рассмотрим алгоритм минимизации продукционного описания блока, реализующего функции переходов автомата.

Шаг 1. Разбиваем систему продукций на группы, в которых все продукции имеют одинаковые правые части.

Шаг 2. В каждой группе с помощью операции 1 все продукции заменяются одной, левая часть которой представляет собой дизъюнкцию левых частей продукций группы.

Шаг 3. В левой части каждой из полученной продукции (в каждой дизъюнктивной нормальной форме) выделяем (если возможно) общий конъюнктивный множитель, который может быть вынесен за скобки. Исключаем этот множитель (в соответствии с операцией 2) из правой части продукции.

Шаг 4. Производим очевидные упрощения левых частей полученных продукций (с помощью обычных операций склеивания и поглощения).

В качестве примера работы описанного алгоритма рассмотрим минимизацию продукционного описания автоматного графа, приведенного на рис. 2. Этот автоматный граф описывает работу автомата без выходов (автомата, выходами которого являются коды состояний). В нашем случае это попросту блок переходов автомата. Этому автоматному графу соответствует автоматная таблица, которая, в свою очередь, может быть описана следующей системой логических продукций:

$$x\bar{s}_1\bar{s}_2 \vdash s_1\bar{s}_2; \quad \bar{x}s_1\bar{s}_2 \vdash s_1s_2;$$

$$xs_1\bar{s}_2 \vdash s_1\bar{s}_2; \quad \bar{x}s_1s_2 \vdash s_1s_2;$$

$$\bar{x}\bar{s}_1\bar{s}_2 \vdash \bar{s}_1s_2; \quad x\bar{s}_1s_2 \vdash s_1s_2;$$

$$\bar{x}\bar{s}_1s_2 \vdash \bar{s}_1s_2; \quad xs_1s_2 \vdash s_1s_2.$$

Как нетрудно видеть в целях простоты вся система продукций уже разбита на группы с одинаковыми правыми частями, т.е. над исходной системой продукций уже выполнен первый шаг описанного выше алгоритма. Всего получилось три группы продукций с правыми частями: $s_1\bar{s}_2$; $\bar{s}_1 s_2$; s_1s_2, т.е. с кодами состояний, в которые имеются переходы.

В результате применения второго шага алгоритма получим систему, состоящую из трех продукций следующего вида:

$$x\bar{s}_1\bar{s}_2 \vee xs_1\bar{s}_2 \vdash s_1\bar{s}_2;$$

$$\bar{x}\bar{s}_1\bar{s}_2 \vee \bar{x}\bar{s}_1s_2 \vdash \bar{s}_1s_2;$$

$$\bar{x}s_1\bar{s}_2 \vee \bar{x}s_1s_2 \vee x\bar{s}_1s_2 \vee xs_1s_2 \vdash s_1s_2.$$

Сформулируем следующее очевидное утверждение.

Утверждение 3. Любой автоматный граф (автомата без выходов) с n состояниями может быть описан системой продукционных правил, состоящей всего из $k \le n$ продукций, где k – число состояний автомата, в каждое из которых имеет место хотя бы один переход.

В соответствии с третьим шагом алгоритма в первой продукции можно выделить общий конъюнктивный множитель $\bar{s}_2$, а во второй – $\bar{s}_1$. В третьей продукции общего конъюнктивного множителя для левой и правой части выделить не удалось. После исключения множителей в первых двух продукциях из их правых частей система продукций примет вид:

$$x\bar{s}_1\bar{s}_2 \vee xs_1\bar{s}_2 \vdash s_1;$$

$$\bar{x}\bar{s}_1\bar{s}_2 \vee \bar{x}\bar{s}_1s_2 \vdash s_2;$$

$$\bar{x}s_1\bar{s}_2 \vee \bar{x}s_1s_2 \vee x\bar{s}_1s_2 \vee xs_1s_2 \vdash s_1s_2.$$

Осталось выполнить последний (четвертый) шаг алгоритма – произвести очевидные упрощения левых частей полученных продукций.

В первых двух продукциях возможно склеивание двух конъюнкций в их левых частях. В третьей – попарное склеивание конъюнкций. В результате окончания работы алгоритма получим следующую простую систему:

$$x\bar{s}_2 \vdash s_1;$$

$$\bar{x}\bar{s}_1 \vdash s_2;$$

$$xs_2 \vee \bar{x}s_1 \vdash s_1s_2.$$

В этой продукционной системе каждая продукция описывает несколько переходов (включая петли) в автоматном графе. Построение автоматного графа по минимизированному продукционному описанию выполняется с применением следующих правил.

Первое правило. Каждая продукция описывает переходы из тех состояний автомата, коды которых содержат кодовые переменные, входящие в левую часть продукции. Например, первая продукция в левой части содержит кодовую переменную $\bar{s}_2$. Следовательно, эта продукция описывает переходы из двух состояний графа на рис. 2 с кодами $\bar{s}_1\bar{s}_2$ и $s_1\bar{s}_2$. Если в левой части продукции отсутствуют кодовые переменные состояний, то такая продукция описывает переходы из всех состояний автоматного графа.

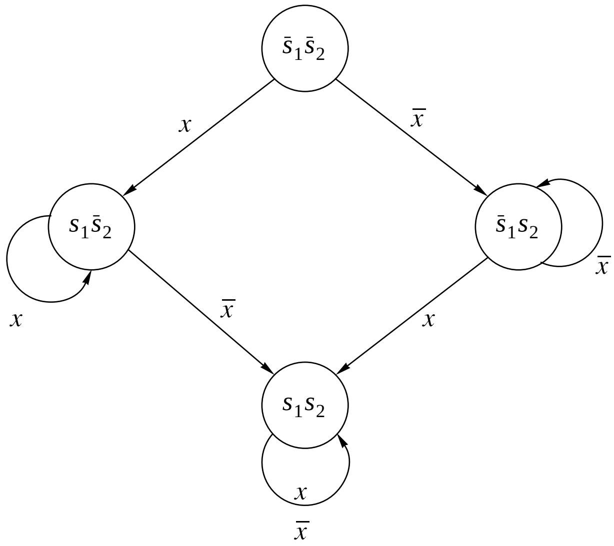

Рис. 2. Граф автомата без выходов, описываемый двумя логическими продукциями: $x \vdash s_1$; $\bar{x} \vdash s_2$.

Второе правило. Код состояния, куда направлен переход, определяется кодом исходного состояния (откуда исходит рассматриваемая дуга) с заменой в нем тех кодовых переменных, которые содержатся в правой части рассматриваемой продукции. Например, если переход, описываемый первой продукцией, исходит из состояния $\bar{s}_1\bar{s}_2$, то соответствующая дуга будет направлена в состояние $s_1\bar{s}_2$. В коде $\bar{s}_1\bar{s}_2$ переменная $\bar{s}_1$ меняется на переменную s_1. Таким образом выполняется построение автоматного графа или автоматной таблицы по минимизированному продукционному описанию.

Рассмотрим теперь вопрос о том, насколько компактным получается продукционное описание, минимизированное с использованием описанного алгоритма. Его особенностью является фиксированный порядок применения двух операций. Если этот порядок изменить, в ряде случаев можно получит более простое решение.

Вернемся к полученному минимизированному продукционному описанию. Применим к третьей продукции правило 1 в обратном порядке, т.е. от одной продукции перейдем к следующим двум:

$$xs_2 \vdash s_1s_2;$$

$$\bar{x}s_1 \vdash s_1s_2.$$

Теперь в каждой полученной продукции появились условия для выполнения операции 2 – исключения из правой части общего конъюнктивного множителя. Проделав это, получим следующие две продукции:

$$xs_2 \vdash s_1;$$

$$\bar{x}s_1 \vdash s_2.$$

Присоединим эти две продукции к оставшимся двум, полученным после реализации алгоритма. Полученная система из четырех продукций может быть разбита на две подсистемы с одинаковыми правыми частями. После применения к ним операции 1 и минимизации левых частей получим предельно простую систему из двух продукций, описывающих автоматный граф, приведенный на рис. 2:

$$x \vdash s_1;$$

$$\bar{x} \vdash s_2.$$

Этот пример говорит о том, что фиксированный порядок применения операций в алгоритме не гарантирует минимального решения. Для поиска минимального решения необходим перебор вариантов применения операций, что свойственно для всего класса задач минимизации булевых функций.

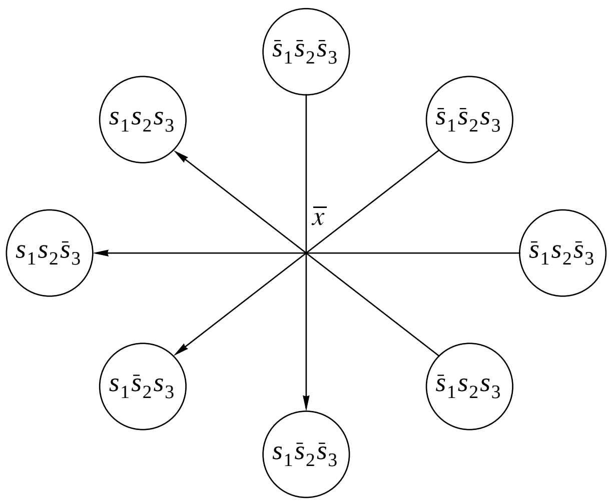

Рис. 3. Фрагмент графа переходов, описываемый единственной логической продукцией: $\bar{x}\bar{s}_1 \vdash s_1$.

Заметим, что если в продукционном описании используются логические продукции, описывающие большое число переходов, продукционное описание в целом получается весьма компактным. На рис. 3 дан пример фрагмента автоматного графа, который описывается единственной продукцией $\bar{x}\bar{s}_1 \vdash s_1$. Особенно удобным язык логических продукций становится в случае сильно недоопределенных автоматов. В этом случае появляются дополнительные возможности минимизации продукционных описаний.

Что же касается минимизации выходного преобразователя автомата, то для решения этой задачи пригодна только операция 1. Именно этой операцией мы воспользовались при минимизации продукционного описания булевой функции $f_1(x_1, x_2, x_3)$, заданной приведенной в разделе 3 таблицей.

Заключение. В работе рассмотрен продукционный подход к описанию исполнительных устройств интеллектуальных систем логического управления. Предложенный язык позволяет формулировать условия работы и строить формальные модели в условиях высокой степени неопределенности используемой информации. С разработкой продукционных языковых средств для исполнительного уровня интеллектуальных систем логического управления становится возможной организация многоуровневых управляющих структур, в которых процессы обработки информации на каждом из уровней могут быть описаны с использованием единственного продукционного подхода. На исполнительном уровне используются логические продукции, на уровне нечеткого управления – нечеткие продукционные системы и так далее вплоть до продукций самого общего вида. Вопросы технической реализации продукционных описаний выходят за рамки данной статьи. Некоторые результаты исследований в этом направлении содержатся в [3], где, в частности, приведена типовая структура аппаратной реализации продукционного описания автомата.

СПИСОК ЛИТЕРАТУРЫ

1. *Захаров В.Н.* Интеллектуальные системы управления. Основные понятия и определения // Изв. РАН. Теория и системы управления. 1997. № 3.
2. *Захаров В.Н., Поспелов Д.А., Хазацкий В.Е.* Системы управления. Задание, проектирование, реализация. М.: Энергия, 1978.
3. *Захаров В.Н.* Автоматы с распределенной памятью. М.: Энергия, 1975.