

УДК 021.8 + 025.1

А. А. Максимов

## ОДИН ПОДХОД К ПОСТРОЕНИЮ КОНЕЧНО-АВТОМАТНОЙ УПРАВЛЯЮЩЕЙ СЕТИ

*Рассмотрена одна из разновидностей конечного автомата с переменной структурой и ее использование как элемента управляющей автоматной сети. Предложен критерий сложности для указанной сети и разработана процедура ее построения. Приведен пример применения разработанного математического аппарата для построения системы логического управления робототехническим комплексом.*

**E-mail:** max2@bmstu.ru

**Ключевые слова:** конечные автоматы, сети Петри, критерий сложности, робототехнический комплекс.

Использование различных разновидностей конечных автоматов для формального описания как объектов управления, так и управляющих устройств, – распространенное явление. Разработан ряд алгоритмов синтеза управляющих автоматов и сетей автоматов (например, [1–2], [4–6]). При этом синтез одного управляющего автомата по формальному описанию объекта управления, заданного также в виде конечного автомата, исследован довольно подробно и не представляет интереса. В случае же, когда нужно построить управляющую автоматную сеть, оптимальную по какому-либо критерию, возникает необходимость полного перебора вариантов (как известно, проблема построения структуры управляющей системы в общем случае NP-полна, т. е. трудоемкость ее решения оценивается экспоненциальной функцией от размерности задачи). Данная работа посвящена разработке алгоритма построения конечно-автоматной управляющей сети, оптимальной по объему занимаемой памяти при ее программной реализации.

Модели объектов и алгоритм их совместного функционирования. Для описания объектов управления будем использовать частично-определенные инициальные автоматы Мили, в общем случае – недетерминированные, на которые наложены отдельные ограничения, касающиеся управляемости и наблюдаемости:

$$A_i = \langle U_i, X_i, Z_i, f_i, h_i, x_0^i \in X_i \rangle, \quad i = 1, \dots, n, \quad (1)$$

где  $f_i : U_i \times X_i \rightarrow 2^{X_i}$  – функция переходов;  $h_i : U_i \times X_i \rightarrow 2^{Z_i}$  – функция выходов;  $x_0^i$  – начальное состояние автомата  $A_i$ .

Ограничения, накладываемые на управляемые автоматы в данной работе, связаны с условием детерминированности синтезируемых управляющих автоматов и подробно изложены в [2], поэтому мы не будем останавливаться на них.

Следует отметить, что и управляющие, и управляемые автоматы могут иметь  $\varepsilon$ -переходы [7], т. е. совершать переходы из состояния в состояние без обработки каких-либо входных символов. Таким образом, в качестве объекта управления задана группа автоматов.

При этом логика совместного функционирования задана безопасной маркированной сетью Петри с ингибиторными дугами (2) с использованием функций привязки параметров управляемых автоматов к параметрам сети Петри (3), (4):

$$C = \langle P, T, I, O, \mu, F_I \rangle, \quad (2)$$

где  $P$  – множество позиций;  $T$  – множество переходов;  $I$  – входная функция;  $O$  – выходная функция,  $\mu$  – функция маркировки,  $F_I$  – функция инцидентности, задающая ингибиторные позиции.

Функции привязки выглядят следующим образом:

$$\Phi : P \rightarrow U_{i=1}^n X_i \quad (3)$$

– привязка позиций сети Петри к состояниям управляемых автоматов;

$$\Psi : T \rightarrow U_{i=1}^n U_i \times Z_i \quad (4)$$

– привязка переходов сети Петри к парам (входной символ, выходной символ) управляемых автоматов.

На сеть Петри, задающую алгоритм совместного функционирования, наложены следующие ограничения, которые так же, как и ограничения на управляемые автоматы, связаны с условием детерминированности синтезируемых управляющих автоматов:

- функции  $I$ ,  $O$  сети Петри и функции  $f_i$  и  $h_i$  управляемых автоматов должны быть согласованы, т. е. последовательность срабатывания переходов множества  $T$  сети Петри должна соответствовать последовательности переходов управляемых автоматов из состояния в состояние;

- при борьбе за ресурс нескольких управляемых автоматов должен быть указан приоритет срабатывания соответствующих переходов сети Петри с помощью ингибиторных дуг.

На рис. 1 представлен фрагмент сети Петри, описывающий борьбу за ресурс автоматов  $A_k$  и  $A_l$ . Значения функции  $\Psi(t_i) = (u^k | z^k)$  и  $\Psi(t_j) = (u^l | z^l)$  показаны на графе рядом с соответствующими переходами. Верхний индекс входных и выходных символов показывает их принадлежность к управляемому автомату, например, выходной символ с номером  $p$  автомата  $A_k$  обозначен как  $z_p^k$ ;

- при альтернативном срабатывании переходов без приоритета (рис. 2) они должны быть приписаны к одному автомату. При этом управляющий символ (на рисунке это  $u^k$ ) должен быть одинаков для всех переходов, имеющих общую входную позицию.

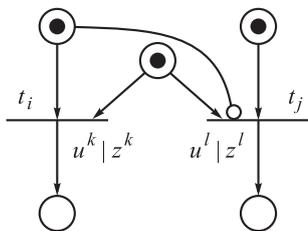


Рис. 1. Задание приоритета перехода  $t_i$  по отношению к  $t_j$

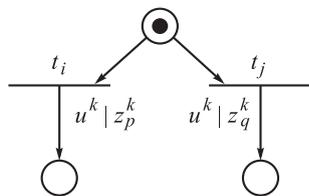


Рис. 2. Альтернативно срабатывающие переходы  $t_i$  и  $t_j$

Таким образом, группа автоматов и закон их функционирования являются исходными данными для построения управляющей сети.

**Управляющий конечный автомат.** В качестве основного элемента управляющей сети будем использовать одну из разновидностей автомата с переменной структурой. Напомним, что автомат с переменной структурой, имеющий  $k$  входов и  $l$  выходов (рис. 3), отличается от классического автомата с несколькими входами и несколькими выходами тем, что в определении его функционирования не учитываются поступающие по некоторым выделенным входным каналам воздействия.

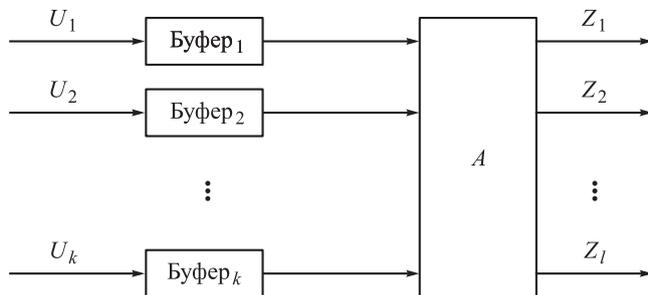


Рис. 3. Конечный автомат с несколькими входами и несколькими выходами

Кроме того, после приема необходимой входной комбинации символов в текущем состоянии данный автомат выдает символы только по выделенным выходным каналам. Такое определение можно найти, например, в работе [3]. Поскольку функционирование управляемых и управляющих автоматов не привязано к единым тактовым моментам времени, во избежание потери комбинаций входных символов каждый вход управляющего автомата дополнен буфером, работающим по принципу FIFO (первым пришел – первым выдан).

Формальное описание такого автомата, имеющего  $k$  входов и  $l$  выходов, выглядит следующим образом:

$$\begin{aligned} R &= \langle U_1, \dots, U_k; X = \{x_1, \dots, x_n\}; Z_1, \dots, Z_l; \\ F &= \{F_1, \dots, F_n\}; H = \{H_1, \dots, H_n\} \rangle, \end{aligned} \quad (5)$$

где  $U_i$  – алфавит, соответствующий  $i$ -му входному каналу,  $i = 1, \dots, k$ ;  $X$  – множество состояний автомата,  $X = \{x_i\}_{i=1}^n$ ;  $Z_i$  – алфавит, соответствующий  $i$ -му выходному каналу,  $i = 1, \dots, l$ ;  $F_i$  – множество частично-определенных функций переходов, связанных с состоянием  $x_i$ ,  $F_i = \{f_1^i, \dots, f_{m_i}^i\}$ ;  $H_i$  – множество частично-определенных вектор-функций выходов, связанных с состоянием  $x_i$ ,  $H_i = \{h_1^i, \dots, h_{m_i}^i\}$ ;

$$\begin{aligned} f_1^i: pr_{(j_1^1, \dots, j_{p_1}^1)} U_1 \times \dots \times U_k &\rightarrow X; \\ f_{m_i}^i: pr_{(j_1^{m_i}, \dots, j_{p_{m_i}}^{m_i})} U_1 \times \dots \times U_k &\rightarrow X; \\ &\vdots \\ h_1^i: pr_{(q_1^1, \dots, q_{l_1}^1)} U_1 \times \dots \times U_k &\rightarrow pr_{(q_1^1, \dots, q_{l_1}^1)} Z_1 \times \dots \times Z_l; \\ h_{m_i}^i: pr_{(q_1^{m_i}, \dots, q_{l_{m_i}}^{m_i})} U_1 \times \dots \times U_k &\rightarrow pr_{(q_1^{m_i}, \dots, q_{l_{m_i}}^{m_i})} Z_1 \times \dots \times Z_l; \end{aligned}$$

$1 \leq i \leq n$ , где  $n$  – число состояний;  $m_i$  – определяет число различных комбинаций выделенных входных каналов, используемых автоматом в состоянии  $x_i$ ,  $1 \leq m_i \leq 2^k$ ;  $j_1^1, \dots, j_{p_1}^1, j_1^2, \dots, j_{p_2}^2, \dots, j_1^{m_i}, \dots, j_{p_{m_i}}^{m_i}$  – номера выделенных входных каналов, используемых автоматом в состоянии  $x_i$ ,

$$\begin{aligned} 1 \leq j_1^1 < \dots < j_{p_1}^1 &\leq k, \\ 1 \leq j_1^2 < \dots < j_{p_2}^2 &\leq k, \\ &\vdots \\ 1 \leq j_1^{m_i} < \dots < j_{p_{m_i}}^{m_i} &\leq k, \end{aligned}$$

$q_1^1, \dots, q_{l_1}^1, q_1^2, \dots, q_{l_2}^2, \dots, q_1^{m_i}, \dots, q_{l_{m_i}}^{m_i}$  – номера выделенных выходных каналов, используемых автоматом в состоянии  $x_i$ ,

$$1 \leq q_1^1 < \dots < q_{l_1}^1 \leq l,$$

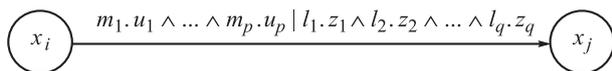
$$1 \leq q_1^2 < \dots < q_{l_2}^2 \leq l$$

⋮

$$1 \leq q_1^{m_i} < \dots < q_{l_{m_i}}^{m_i} \leq l.$$

Очевидно, что символы, поступающие на входы автомата (5) по каналам, на которые он не реагирует в текущем состоянии, являются фиктивными параметрами для функций  $f$  и  $h$  и, следовательно, для определения его функционирования могут не учитываться, что значительно сокращает область определения названных функций.

Для задания автомата с переменной структурой удобно использовать диаграмму Мура с некоторым дополнением, определяющим номера выделенных входных и выходных каналов. Так, если автомат переходит из состояния  $x_i$  в состояние  $x_j$  под воздействием некоторой входной комбинации символов  $(u_1, u_2, \dots, u_p)$  (рис. 4), то перед каждым символом указывается номер выделенного входного канала, по которому этот символ поступает.



**Рис. 4. Фрагмент модифицированной диаграммы Мура**

При этом номер входного канала и символ разделяются точкой. Если автомат выдает при переходе из состояния в состояние выходную комбинацию  $(z_1, \dots, z_q)$ , то аналогично входной комбинации каждый выходной символ следует за номером выходного выделенного канала. Символ и номер выходного канала также разделяются точкой. Таким образом, пара  $m_i.u_i$ ,  $i = 1, \dots, p$ , означает, что символ  $u_i$  поступает на вход  $m_i$ , а пара  $l_i.z_i$ ,  $i = 1, \dots, q$ , – что символ  $z_i$  выдается на выходной канал  $l_i$ .

**Критерий сложности.** В работе [8] указывается, что основными характеристиками сложности являются размерность, неопределенность и информационно-структурные ограничения. В качестве критерия, по которому будет строиться сеть, в данной работе выбрана размерность. Соответствующая ей целевая функция имеет следующий вид:

$$D = \alpha N_F + \beta N_R + \gamma N_C \rightarrow \min, \quad (6)$$

где  $N_F$  – суммарная сложность всех управляющих автоматов, которая тесно связана с размерностью и содержанием таблиц, задающих функции переходов и выходов;  $N_R$  – количество управляющих автоматов;  $N_C$  – общее число связей между автоматами сети (как управляемыми, так и управляющими);  $\alpha, \beta, \gamma$  – весовые коэффициенты, определяющие значимость каждого из параметров.

В данной работе рассматривается алгоритм построения сети, обеспечивающий минимум целевой функции (6).

При программной реализации управляющей автоматной сети указанный критерий определяет требуемый объем оперативной памяти вычислительной системы. В этом случае значения весовых коэффициентов надо поставить в соответствие с размерами элементов структур данных, сформированных разработчиком для задания параметров автоматной сети. При этом реальное значение критерия напрямую зависит от искусства программиста экономно использовать память при написании программного кода, а также от применяемой операционной системы.

Рассмотрим подробнее параметр  $N_F$  критерия (6). Определим сложность состояния  $x_i$  автомата как суммарную сложность всех отношений ⟨входная комбинация, выходная комбинация⟩, приписанных указанному состоянию, и обозначим ее как  $||x_i||$ . Сложность отношения определим как количество пар ⟨номер входа, символ⟩ и ⟨номер выхода, символ⟩, составляющих все входные и выходные комбинации, которые связаны с данным состоянием. Сложность конечного автомата  $A$  (обозначим ее как  $||A||$ ) определим как суммарную сложность всех его состояний, т.е.

$$||A|| = \sum_{i=1}^{m_A} ||x_i||, \quad (7)$$

где  $m_A$  – количество состояний автомата.

Тогда суммарная сложность всех управляющих автоматов будет определяться следующим образом:

$$N_F = \sum_{\substack{1 \leq i \leq m_R \\ 1 \leq j \leq N_R}} ||x_i^j||, \quad (8)$$

где  $x_i^j$  – состояние с номером  $i$ , принадлежащее автомату  $R_j$ .

**Алгоритм построения управляющей сети.** Для поиска оптимального решения необходимо сравнивать значения целевой функ-

ции (6) для всех вариантов, каждый из которых подразумевает следующие этапы:

- выбор структуры автоматной сети;
- построение функций переходов и выходов управляющих автоматов;
- оценка сложности сети по критерию (6).

Выбор структуры автоматной сети тесно связан с разбиением переходов множества  $T$  на подмножества. Каждый переход сети Петри  $t \in T$  связан посредством функции привязки  $\Psi$  с переходом конкретного управляемого автомата из состояния в состояние и может быть «поручен» одному из управляющих автоматов. Разбивая множество  $T$  на подмножества, ставя в соответствие этим подмножествам свои управляющие автоматы и определяя связи между ними, мы тем самым задаем структуру управляющей автоматной сети.

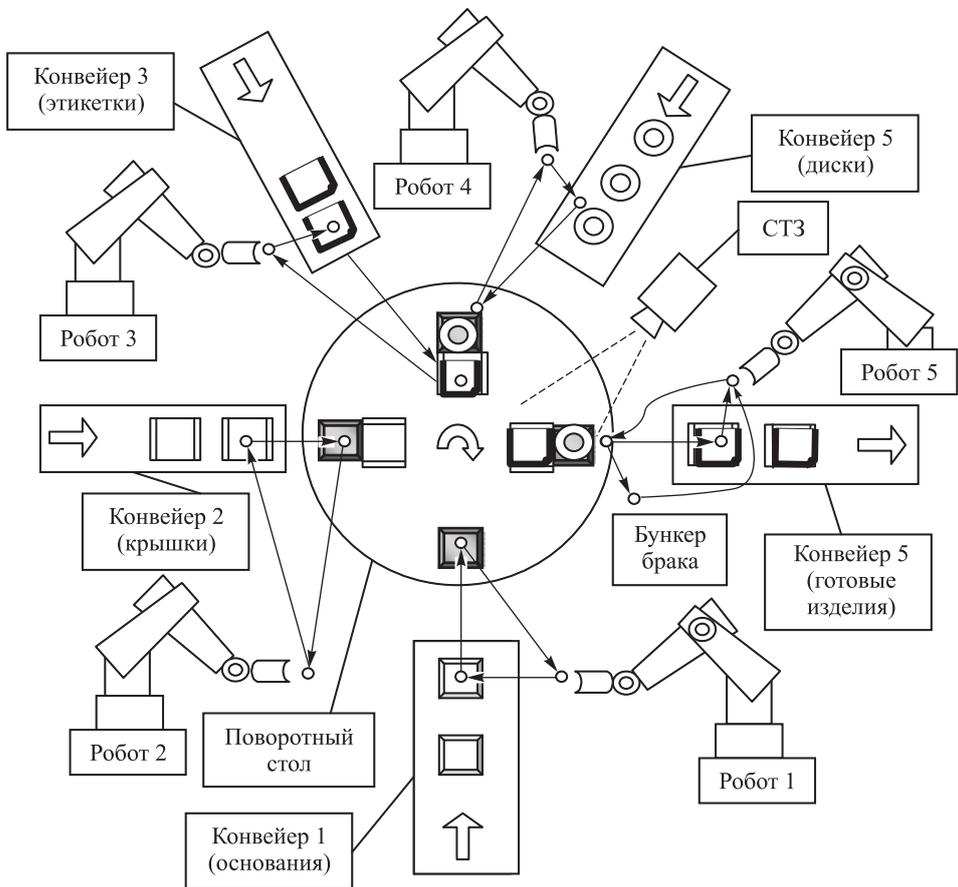
Пусть  $T = \bigcup_{i=1}^w T_i$  – разбиение множества  $T$ , т. е.  $T_j \cap T_k = \emptyset$  при  $1 \leq j < k \leq w$ . Каждый блок разбиения  $T_i$  поставим в соответствие управляющему автомату  $R_i$ . Следует отметить, что при определенном соотношении весовых коэффициентов оптимальная по критерию (6) сеть может содержать, кроме управляющих автоматов, непосредственно связанных с управляемыми, автоматы, координирующие работу самих управляющих автоматов. Будем обозначать такие автоматы следующим образом:  $R_j^R$ ,  $j=1, \dots, v$ , где  $v$  – число указанных автоматов. Введем также следующее обозначение:  $[[T_i]]$  – сложность автомата, порождаемая блоком разбиения  $T_i$ , т. е. сложность  $R_i$ . Тогда критерий (6) можно записать в следующем виде:

$$D\left(\bigcup_{i=1}^w T_i\right) = \alpha \left( \sum_{i=1}^w [[T_i]] + \sum_{j=1}^v [[R_j^R]] \right) + \beta(w+v) + \gamma N_C. \quad (9)$$

Запись  $D\left(\bigcup_{i=1}^w T_i\right)$  показывает, что значение критерия непосредственно зависит от разбиения множества  $T = \bigcup_{i=1}^w T_i$ .

Приведем в качестве примера построение одного из вариантов управляющей автоматной сети для сборочного робототехнического комплекса (РТК), предназначенного (рис. 5) для сборки кассет, состоящих из основания, крышки, этикетки и оптического диска.

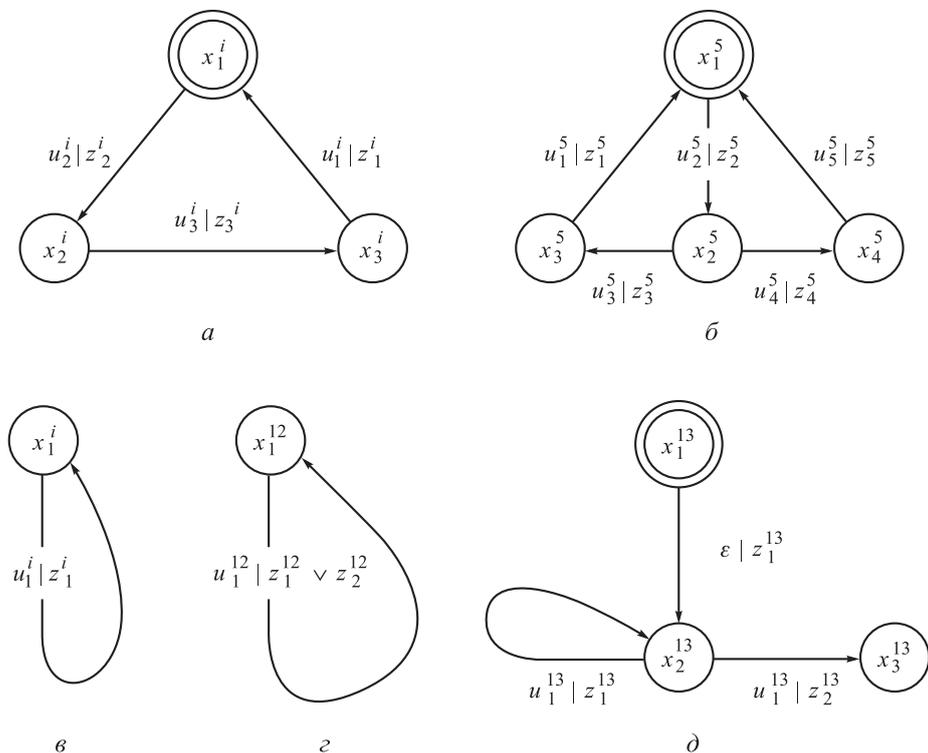
Указанные комплектующие подаются на место сборки с помощью шаговых конвейеров; сборка изделия производится на поворотном столе пятью роботами. Последовательность сборки выглядит следующим образом:



**Рис. 5. Сборочный робототехнический комплекс**

- системой управления верхнего уровня подается команда о начале сборки;
- осуществляется поворот стола на  $90^\circ$  по часовой стрелке;
- роботы 1–4 перемещают соответствующие детали с конвейеров на сборочные позиции поворотного стола. При этом невозможно одновременное нахождение в зоне сборки роботов 3 и 4 (приоритет имеет робот 3). Параллельно осуществляется проверка правильности сборки предыдущего изделия системой технического зрения (СТЗ). В случае брака изделие попадает в специальный бункер; при правильной сборке робот 5 перемещает его на конвейер готовой продукции;
- каждый из шаговых конвейеров 1–4 подает следующую деталь на соответствующую рабочую позицию. Конвейер 5 с готовой продукцией срабатывает в том случае, если предыдущая деталь была собрана правильно и не попала в бункер брака;
- осуществляется запрос системы управления верхнего уровня, которая либо инициирует сборку очередного изделия, либо завершает работу РТК.

Модели объектов управления в виде конечных автоматов представлены на рис. 6.



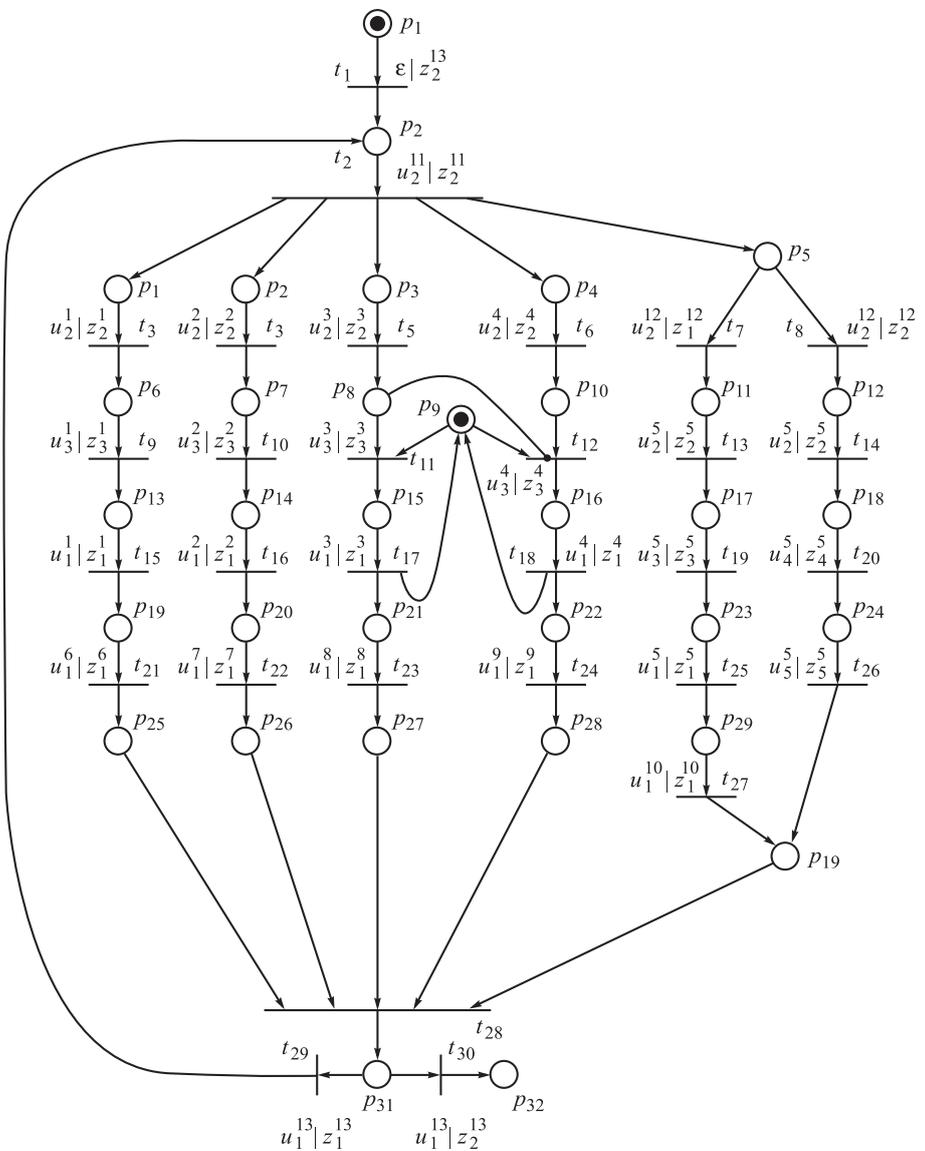
**Рис. 6. Модели объектов:**

*a* –  $A_i$  – модель робота  $i$ ,  $i=1, \dots, 4$ ; *б* –  $A_5$  – модель робота 5; *в* –  $A_i$ ,  $i=6, \dots, 10$ , – модель конвейера с номером  $(i-5)$ ;  $i=11$  – модель поворотного стола; *г* –  $A_{12}$  – модель СТЗ; *д* –  $A_{13}$  – модель системы управления верхнего уровня

Алгоритм совместной работы указанных автоматов задан сетью Петри (рис. 7) с помощью функций привязки. При этом значения функции привязки  $\Psi$  показаны в виде пар  $u_p^k | z_p^k$  рядом с переходами сети Петри. Значения функции  $\Phi$  не показаны рядом с позициями сети Петри для того, чтобы не загромождать рисунок. Рассмотрим одно из разбиений переходов и построим структуру управляющей автоматной сети на основе данного разбиения.

Пусть  $T = \bigcup_{i=1}^5 T_i$ , где блоки разбиения выглядят следующим обра-

зом:  $T_1 = \{t_1, t_2, t_3, t_9, t_{15}, t_{21}, t_{28}, t_{29}, t_{30}\}$ ,  $T_2 = \{t_4, t_{10}, t_{16}, t_{22}\}$ ,  $T_3 = \{t_5, t_{11}, t_{17}, t_{23}\}$ ,  $T_4 = \{t_6, t_{12}, t_{18}, t_{24}\}$ ,  $T_5 = \{t_7, t_8, t_{13}, t_{14}, t_{19}, t_{20}, t_{25}, t_{26}, t_{27}\}$ .



**Рис. 7. Алгоритм совместной работы конечных автоматов**

Каждому блоку разбиения  $T_i$ , как указывалось ранее, ставится в соответствие управляющий автомат  $R_i$ . На основании указанного разбиения строятся связи между управляющими и управляемыми автоматами следующим образом:

- управляемый автомат  $A_j$  связан входом и выходом с управляющим автоматом  $R_i$  в том случае, если блоку разбиения  $T_i$  принадлежит хотя бы один переход  $t_k$  такой, что  $\Psi(t_k) = u_p^j | z_q^j$ , т. е. переход сети Петри, соответствующий переходу автомата  $A_j$  из состояния в состояние;

• управляющий автомат  $R_i$  связан с управляющим автоматом  $R_j$ , если выполнено одно из двух условий:

а) существует позиция сети Петри, являющаяся одновременно выходной для одного из переходов блока разбиения  $T_i$  и входной для одного из переходов блока  $T_j$ ;

б) существует простой путь в сети Петри между каким-либо переходом блока разбиения  $T_i$  и переходом блока  $T_j$ , содержащий только те переходы, на которых значение функции  $\Psi$  не определено (такая ситуация возможна в случае использования вспомогательных переходов при задании логики совместной работы управляемых автоматов, т. е. функция  $\Psi$  в общем случае может быть частично определена).

Структура управляющей сети, соответствующая выбранному разбиению, показана на рис. 8, диаграммы управляющих автоматов  $R_1 - R_6$  приведены на рис. 9 и 10. Для построения автоматов  $R_1 - R_5$  использован аналог алгоритма, подробно рассмотренного в [2] и основанного на построении формальной системы, описывающей возможные вход-выходные последовательности автоматов, которые соответствуют переходам сети Петри согласно функции  $\Psi(t_k)$ .

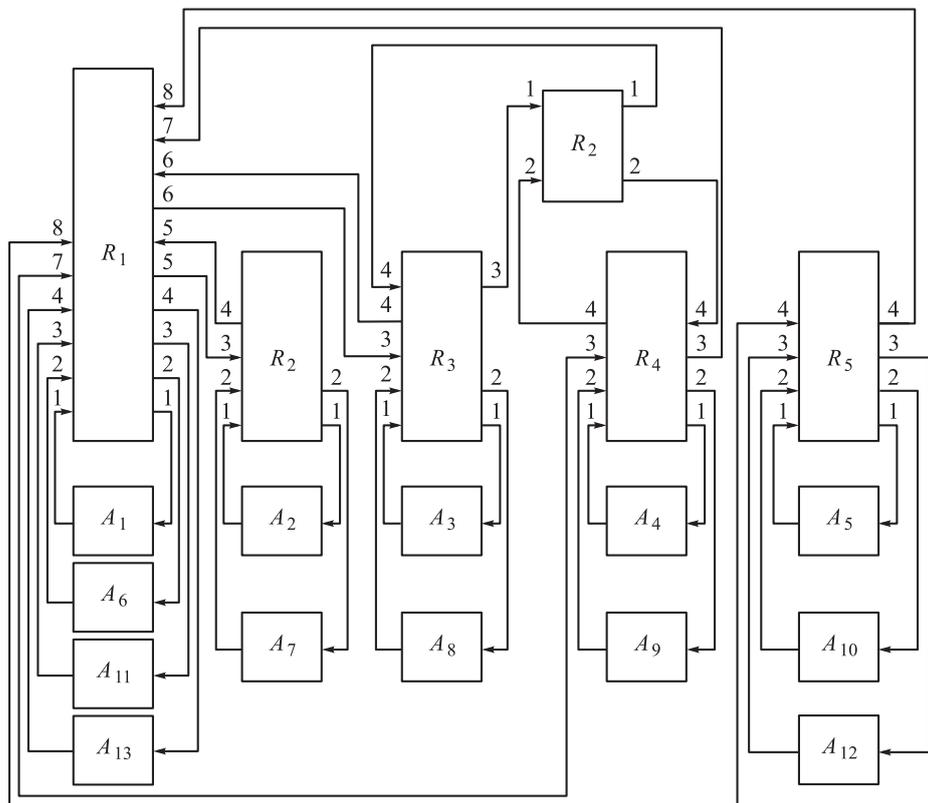
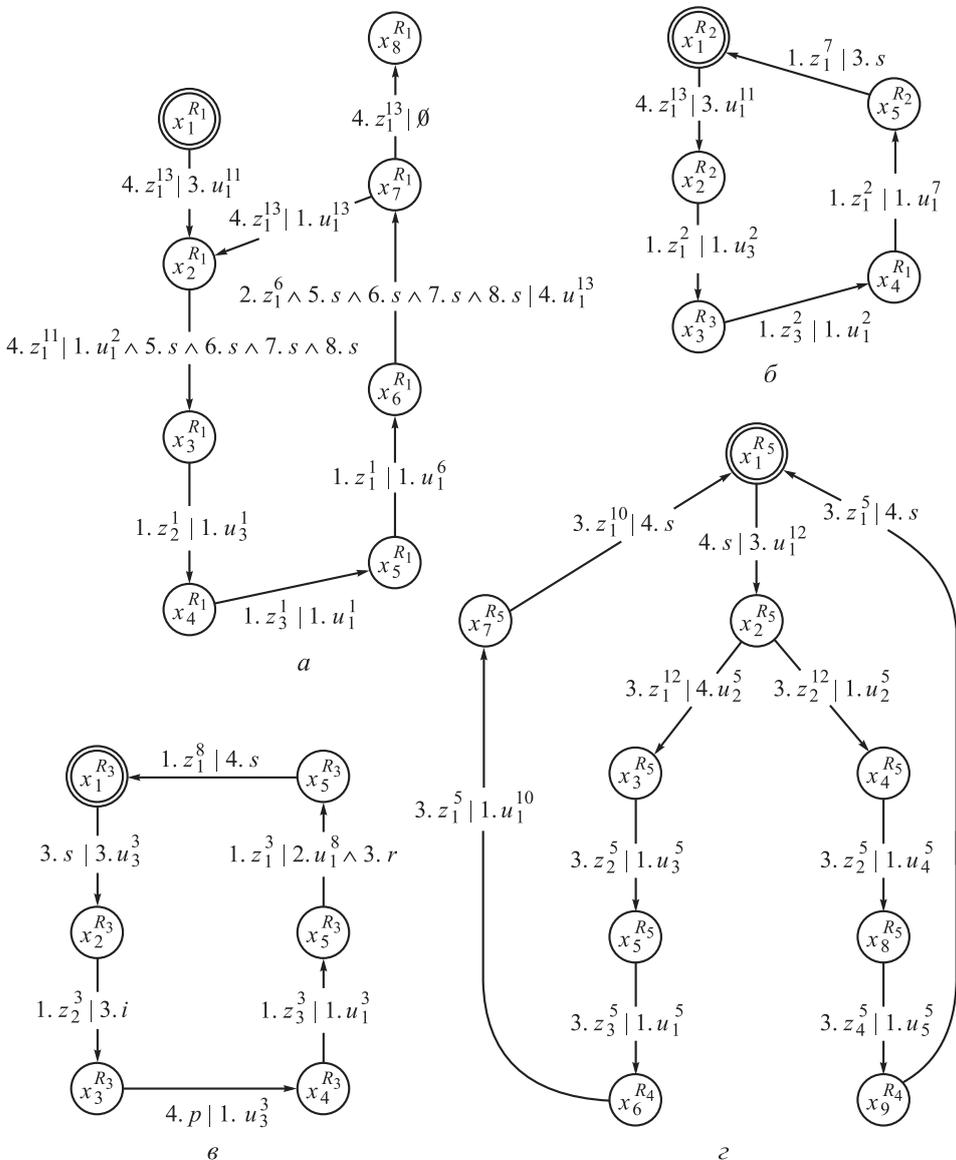


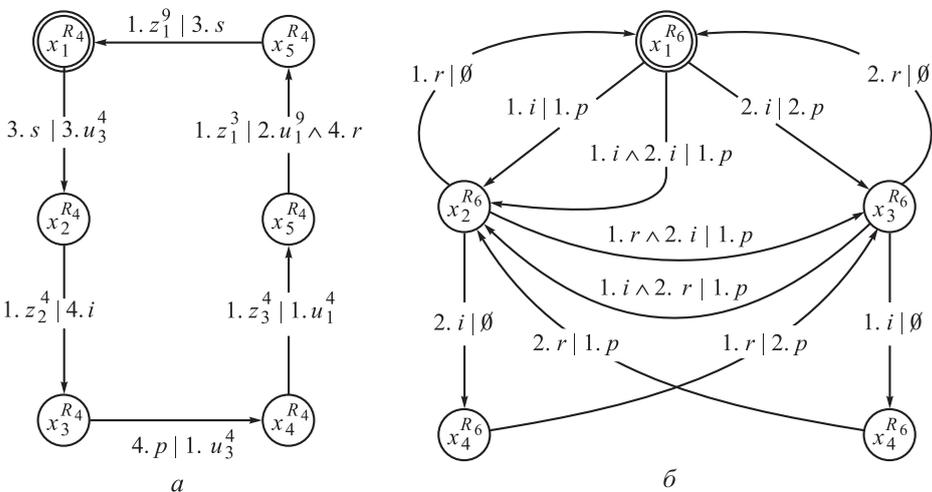
Рис. 8. Структура управляющей автоматной сети



**Рис. 9. Управляющие автоматы:**

*a* –  $R_1$ ; *б* –  $R_2$ ; *в* –  $R_3$ ; *г* –  $R_5$

Кроме управляющих автоматов, соответствующих блокам разбиения  $T_1, \dots, T_5$ , в данной сети присутствует автомат  $R_6^R$  (см. рис. 8), обеспечивающий корректную борьбу за ресурс автоматов  $R_3$  и  $R_4$ . Для построения такого автомата существует несложный алгоритм, подробное изложение которого приведено в [4]. Отметим, что для задания автомата, распределяющего ресурс между автоматами согласно приоритету, используются следующие символы: *i* – запрос ресурса (inquiry), *r* – отказ от ресурса (refuse), *p* – предоставление ресурса (provide).



**Рис. 10. Управляющие автоматы:**

$a - R_4$ ;  $b - R_6^R$

Далее проводится оценка сложности построенного варианта управляющей автоматной сети по критерию сложности (9):

1) сложность управляющих автоматов:

$$|[R_1]| = 24, |[R_2]| = 10, |[R_3]| = 13, |[R_4]| = 13, |[R_5]| = 20, |[R_6^R]| = 25.$$

Тогда суммарная сложность управляющих автоматов:

$$\sum_{i=1}^5 |[R_i]| + |[R_6^R]| = 24 + 10 + 13 + 13 + 20 + 25 = 105 - \text{такое количество пар } \langle \text{номер входа, символ} \rangle \text{ и } \langle \text{номер выхода, символ} \rangle \text{ содержат все управляющие автоматы сети;}$$

2) общее число управляющих автоматов:  $N_R = 6$ ;

3) общее число связей между управляющими и управляемыми автоматами, т.е. число пар  $\langle \text{выход, вход} \rangle$ :  $N_C = 38$ .

Тогда для данного варианта управляющей сети значение критерия выглядит следующим образом:

$$D\left(\bigcup_{i=1}^5 T_i\right) = \alpha \left( \sum_{i=1}^5 |[R_i]| + |[R_6^R]| \right) + \beta(5+1) + \gamma N_C = \alpha \cdot 105 + \beta 6 + \gamma 38.$$

В качестве примера проведем оценку объема оперативной памяти в байтах, занимаемой данной автоматной сетью при ее программной реализации в операционной системе реального времени QNX v.6.0. Пусть элемент структуры данных, введенный программистом для

хранения номера входа или выхода конечного автомата, занимает 2 байта, а элемент для хранения символа – 4 байта. Тогда, как указывалось ранее при определении критерия (6), для хранения пары  $\langle \text{номер входа}, \text{символ} \rangle$  или пары  $\langle \text{номер выхода}, \text{символ} \rangle$  потребуется суммарно 6 байт, т. е. весовой коэффициент  $\alpha$  равен 6. Предположим, что для каждого управляющего автомата в операционной системе выделен отдельный поток, которому для работы необходимо 2000 байт оперативной памяти. Тогда весовой коэффициент  $\beta$  равен 2000. Предположим также, что для хранения идентификатора, уникального для каждого автомата, заведено 6 байт. Тогда для хранения следующей тетрады параметров:

(идентификатор автомата, номер выхода,  
номер входа, идентификатор автомата),

определяющей одну связь между автоматами, потребуется 16 байт, т. е. коэффициент  $\gamma$  равен 16. В результате объем памяти, занимаемый построенной автоматной сетью, будет следующим:

$$\alpha \cdot 105 + \beta \cdot 6 + \gamma \cdot 38 = 6 \cdot 105 + 2000 \cdot 6 + 16 \cdot 38 = 13\,238 \text{ байт.}$$

Количество перебираемых и оцениваемых по критерию (6) вариантов, подлежащих сравнению для выбора оптимального, ограничено сверху числом Белла  $B_{|T|}$ , т.е. числом всех разбиений множества переходов сети Петри. В нашем случае – это  $B_{30}$ , поскольку  $|T| = 30$ . Это довольно значительное число, например:

$$B_{10} = 115\,975,$$

$$B_{20} = 51\,724\,158\,235\,372.$$

Поэтому на практике можно найти способы уменьшения перебираемых вариантов и понизить ограничение их количества до числа Стирлинга второго рода  $S(|T|, w)$ , где  $w$  – число блоков разбиения. Так, при проведенном заранее анализе сети Петри можно установить максимальное количество независимо работающих автоматов и (при определенном соотношении весовых коэффициентов  $\alpha, \beta, \gamma$ ) рассматривать только те разбиения множества  $T$ , каждый блок которых не содержит внутри независимо срабатывающих переходов. Под независимыми будем понимать переходы, способные при какой-либо достижимой маркировке сработать без взаимного исключения. Однако подробное рассмотрение способов сокращения перебираемых вариантов выходит за рамки данной работы.

В заключение следует отметить, что в Научно-учебном центре «Робототехника» МГТУ им. Н.Э. Баумана разработан пакет программ LOCOL, позволяющий автоматизировать процесс получения оптимальной управляющей сети, который включает графические редакторы для задания как моделей объектов, так и алгоритма их функционирования в виде сети Петри. Кроме того, пакет включает программные модули, позволяющие привязать символы конечных автоматов к данным и запустить автоматную сеть на исполнение, обеспечивая управление логикой работы сложных систем с использованием стандартных интерфейсов.

## СПИСОК ЛИТЕРАТУРЫ

1. Зенкевич С. Л., Ющенко А. С. Основы управления манипуляционными роботами. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004.
2. Иванов Н. Н., Михайлов Г. И., Руднев В. В., Таль А. А. Конечные автоматы: эквивалентность и поведение. – М.: Наука, 1984.
3. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. – М.: Наука, 1985.
4. Максимов А. А., Гао Чжинин. Управление робототехническими комплексами на основе конечных автоматов с переменной структурой // Вестник МГТУ им. Н.Э. Баумана. 2001. № 1.
5. Новорусский В. В. Конечнoавтоматные системы управления (принципы построения и анализ поведения). – Новосибирск.: Наука, 1982.
6. Руднев В. В. Конечный автомат как объект управления // Автоматика и телемеханика. – 1978. № 9.
7. Рейурд-Смит В. Дж. Теория формальных языков. – М.: Радио и связь, 1988.
8. Шильяк Д. Децентрализованное управление сложными системами. – М.: Мир, 1994.

Статья поступила в редакцию 28.06.2012