

УДК 681.32

МИНИМИЗАЦИЯ ГРАФА ДОСТИЖИМЫХ СОСТОЯНИЙ ПАРАЛЛЕЛЬНОГО АВТОМАТА



Л.Д. Черемисинова

*Главный научный сотрудник ОИПИ НАНБ
доктор технических наук, профессор
cld@newman.bas-net.by*

Л. Д. Черемисинова

Окончила Томский государственный университет, доктор технических наук, профессор. Работает в ОИПИ НАН Беларуси в должности главного научного сотрудника и Белорусском государственном университете информатики и радиоэлектроники в должности профессора. Круг научных интересов: дискретная математика, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Аннотация. Рассматривается проблема тестирования управляющих систем, которым присущ параллелизм происходящих в них процессов. В качестве языка задания таких систем используется модель параллельного автомата, представляющая собой подкласс цветных сетей Петри. Анализ поведенческих свойств параллельного автомата проводится на основе его графа достижимости, вершинам которого соответствуют все возможные разметки состояний, число которых растет экспоненциально с ростом степени параллельности состояний. В работе предлагается метод минимизации графа достижимости путем сокращения числа его вершин на основе введения доопределения частичного порядка на множестве параллельно срабатывающих переходов автомата.

Ключевые слова: параллельный автомат, граф достижимых состояний, тестирование дискретных устройств.

Введение.

Развитие микроэлектроники и средств автоматизации проектирования привели к возможности проектирования микроэлектронных управляющих систем значительной сложности. В связи с этим первостепенное значение получила проблема обеспечения надежности и предсказуемости поведения таких сложных систем. Неотъемлемой частью процесса проектирования стало тестирование, в частности, проверка соответствия поведения устройства требованиям, предъявляемым спецификацией на его проектирование. Сложность тестирования возрастает быстрее, чем сложность тестируемых систем, в том числе и в связи с растущими требованиями к качеству [1, 2].

Под тестированием здесь понимается проверка схемной реализации на вход-выходное соответствие модели (input-output conformance), которая состоит в

- генерации проверяющей последовательности входных сигналов на основе заданной спецификации;
- моделировании схемной реализации устройства на полученной тестовой последовательности;
- наблюдении реакций на подаваемые тестовые последовательности сигналов;
- определении, соответствуют ли вход-выходные реакции схемной реализации спецификации на ее проектирование.

Можно выделить следующие два основных способа генерации тестовых входных воздействий (для проверяющей последовательности):

- генерация псевдослучайных наборов воздействий;

– генерация наборов воздействий и проверяющей последовательности в целом, исходя из описания модели.

Минусы первого подхода заключаются в том, что число возможных входных воздействий в каждом состоянии схемы (и модели) экспоненциально зависит от числа переменных, и, главное, не все такие наборы попадают в область определения модели, а значит, и штатного поведения схемной реализации. Второй подход обеспечивает не только наиболее полное тестирование схемы на области, определяемой спецификацией на ее проектирование, но и позволяет сократить длину проверяющей последовательности.

В настоящее время наиболее разработанным направлением в области тестирования является верификация на основе моделей, представляемых конечными автоматами [3, 4]. Менее изучена проблема тестирования систем, которым присущ параллелизм происходящих в них процессов [5, 6]. К ним относятся системы, обеспечивающие управление взаимодействующими компонентами, которые работают параллельно и асинхронно. Параллелизм, присутствующий в объектах управления, отражается в функциональной модели цифровых систем, управляющих данными объектами. Обычно рассматриваются системы, которые состоят из параллельно работающих компонентов и моделируются сетями конечных автоматов. Менее изучена задача тестирования систем с использованием моделей «истинного параллелизма», таких как сети Петри [7, 8]. В таких системах процессы выполняются параллельно и независимо, и нет необходимости контролировать порядок их выполнения. Для управляющих систем рассматриваемого класса характерно также и то, что управляющие воздействия и сигналы о состоянии объектов управления описываются булевыми переменными, лишь небольшой процент всей информации является числовым. Основным инструментом анализа поведенческих свойств моделей этого класса является граф достижимых состояний, в процессе его обхода генерируется тестовая последовательность [7, 8, 9].

В настоящей работе рассматривается задача тестирования систем управления с «истинным параллелизмом» происходящих в них процессов. В качестве языка задания спецификации используется язык параллельных алгоритмов логического управления, а точнее задание описаний в стандартном виде – моделью, названной параллельным автоматом [10]. Алгоритмы в таком виде представляют собой подкласс цветных сетей Петри – расширенные сети свободного выбора [11, 12]. Тестовая последовательность для анализа схемной реализации параллельного автомата строится в процессе обхода его графа достижимости, вершинам которого соответствуют все возможные достижимые состояния автомата.

Построение графа достижимости (в явном или неявном виде) всегда возможно для моделей с конечным пространством состояний, однако проблемой является экспоненциальный рост числа вершин графа достижимости. Основной причиной такого роста (кроме большого числа состояний) является наличие параллельно происходящих в исследуемой системе процессов, что отражается в степени параллелизма состояний. Большое пространство состояний негативно влияет на время тестирования. Более того, параллельный автомат может быть настолько сложным, что становится невозможным в приемлемое для практики время построить граф достижимости и получить даже какую-нибудь тестовую последовательность путем обхода его вершин.

В настоящей работе описываются пути сокращения графа достижимости на основе введения отношения частичного порядка на множестве параллельно срабатывающих переходов. Предлагается модификация метода построения графа достижимости параллельного автомата, предложенного в работе [13]. Целью модификация является сокращение числа вершин графа, в основе которого лежит частичное упорядочение пар переходов автомата на основе учета их информационного взаимодействия.

Параллельный автомат. Параллельный автомат можно рассматривать как динамическую дискретную систему, допускающую параллельно протекающие процессы, которым соответствуют параллельные частичные состояния. В параллельном автомате [10] (в

отличие от классического конечного автомата) рассматриваются частичные состояния, названные так потому, что в любой момент времени параллельный автомат может одновременно находиться в нескольких таких состояниях. Множество частичных состояний в момент времени определяет полное (глобальное) состояние S_t параллельного автомата, которое можно интерпретировать как маркировку сети Петри. Частичные состояния, составляющие полное состояние автомата, соответствуют параллельно протекающим процессам и называются параллельными. Следует заметить, что количество полных состояний параллельного автомата растет экспоненциально с ростом числа частичных состояний и переходов соответствующего параллельного автомата [10].

Параллельный автомат задается совокупностью множеств $\{S, X, Y, T\}$ частичных состояний, входных переменных, выходных переменных и переходов между подмножествами частичных состояний $S_{i1}, S_{i2} \subseteq S$. Переход параллельного автомата задается в форме

$$\tau_i = (\gamma_i^1, k_i^1) \rightarrow (\gamma_i^2, k_i^2),$$

где γ_i^1 и γ_i^2 трактуются как подмножества частичных состояний из S , в которых автомат находится перед и после срабатывания i -го перехода; k_i^1 и k_i^2 – элементарные конъюнкции входных и выходных булевых переменных, трактуемые как условие перехода и выходные сигналы, сопровождающие переход.

Переход срабатывает, когда текущее полное состояние S_t автомата включает все частичные состояния из γ_i^1 ($\gamma_i^1 \subseteq S_t$) а переменные автомата принимают значения, обращающие элементарную конъюнкцию k_i^1 в единицу ($k_i^1 \wedge W_t = 1$) на множестве W_t значений логических переменных системы управления, которое было достигнуто к моменту времени t . Таким образом, динамика параллельных автоматов описывается в пространстве достижимых полных состояний и значений логических сигналов.

Переходы параллельного автомата могут происходить последовательно или параллельно, в последнем случае в случае корректного автомата (рассматриваются только они) для любой пары переходов τ_i и τ_j ($i \neq j$) выполняются условия: если $\gamma_i^1 \cap \gamma_j^1 \neq \emptyset$, то $\gamma_i^1 = \gamma_j^1$, и, если переходы τ_i и τ_j параллельны, то $\gamma_i^2 \neq \gamma_j^2$.

Чтобы вычислить следующее полное состояние S_{t+1} автомата, необходимо определить множество T переходов τ_i , которые срабатывают в состоянии S_t и состоянии W_t на множестве переменных автомата. Следующие состояния S_{t+1} и W_{t+1} задаются следующим образом:

1) $S_{t+1} = (S_t \setminus \gamma^1) \cup (\gamma^2)$, где $\gamma^1 = \cup_i \gamma_i^1$ и $\gamma^2 = \cup_i \gamma_i^2$ есть объединения множеств частичных состояний всех параллельно выполняемых переходов $\tau_i \in T$;

2) состояние W_{t+1} на множестве переменных получается из W_t таким изменением значений переменных, входящих в конъюнкции k_i^2 всех переходов $\tau_i \in T$, что все $k_i^2 = 1$.

Ниже приведен пример описания простого параллельного автомата из работы [13], определенного на множествах $S = \{1, 2, \dots, 11\}$ частичных состояний, $X = \{x_1, x_2\}$ и $Y = \{y_1, y_2\}$ входных и выходных (управляющих) переменных и девяти переходах:

$$\tau_1 = (1, x_1 \underline{x_2}) \rightarrow (10, y_1 \underline{y_2});$$

$$\tau_2 = (10, \underline{x_2}) \rightarrow (2.3.4, -);$$

$$\tau_3 = (2, -) \rightarrow (5.6, y_1);$$

$$\tau_4 = (3.5, \underline{x_2}) \rightarrow (8, -);$$

$$\tau_5 = (4, x_1) \rightarrow (7, y_1);$$

$$\tau_6 = (4, x_1) \rightarrow (9, y_2);$$

$$\tau_7 = (7, \underline{x_2}) \rightarrow (9, -);$$

$$\tau_8 = (6.8.9, -) \rightarrow (11, \underline{y_2});$$

$$\tau_9 = (11, x_1) \rightarrow (1, -).$$

Генерация графа достижимых состояний. Граф достижимости автомата представляет собой ориентированный граф, вершинам которого соответствуют все возможные полные состояния P_i автомата, а дугам – переходы между этими состояниями. Дуга графа помечается

символом перехода τ_i и связывает вершины графа, помеченные множествами полных состояний P_p и P_q , если результат срабатывания перехода τ_i меняет состояние P_p автомата на P_q . Граф достижимых состояний представляет собой в общем случае мультиграф, в котором могут быть петли и кратные дуги, различающиеся присвоенными им метками. Кроме того, очевидно, что каждый переход алгоритма управления может повторяться много раз в качестве метки разных дуг графа достижимости.

Граф достижимости получается путем вычисления всех полных состояний P_i (начиная с начальной), которые ставятся в соответствие вершинам графа, и дуг, связывающих вершины, которые помечаются теми переходами автомата, которые вызывают изменение соответствующих полных состояний [13].

Множество полных состояний $P = \{P_1, P_2, \dots, P_l\}$ автомата строится последовательно. Начиная с заданного начального состояния $P = \{P_1\}$, выбирается очередное уже сформированное полное состояние $P_i \subseteq P$, и для него производится последовательный просмотр всех переходов τ_j автомата. Если для текущего выбранного перехода $\tau_j = (\gamma_j^1, k_j^1) \rightarrow (\gamma_j^2, k_j^2)$ выполняется $\gamma_j^1 \subseteq P_i$, множество полных состояний P заменяется на $P \cup \{(P_i \setminus \gamma_j^1) \cup \gamma_j^2\}$ (увеличивается в общем случае). Процесс заканчивается после просмотра всех состояний из P .

В результате работы алгоритма для приведенного выше параллельного автомата получается множество, состоящее из 12 полных состояний: $\{\{1\}, \{10\}, \{2, 3, 4\}, \{3, 4, 5, 6\}, \{2, 3, 7\}, \{2, 3, 9\}, \{4, 6, 8\}, \{3, 5, 6, 7\}, \{6, 8, 7\}, \{3, 5, 6, 9\}, \{6, 8, 9\}, \{11\}\}$.

Найденные полные состояния P_1, P_2, \dots, P_l являются вершинами ориентированного графа достижимых состояний. Из вершины P_g исходит дуга, заходящая в вершину P_h , если в исходном задании имеется переход $\tau_j = (\gamma_j^1, k_j^1) \rightarrow (\gamma_j^2, k_j^2)$, такой, что $P_h = (P_g \setminus \gamma_j^1) \cup \gamma_j^2$. Метод построения графа достижимости, лежащий в основе данного алгоритма [13], заключается в переборе пар полных состояний (P_g, P_h) и поиске перехода τ_j переводящего P_g в P_h или P_h в P_g . Из вершины P_g графа достижимости исходит дуга в вершину P_h , если $P_g \setminus P_h = \gamma_j^1$ и $P_h \setminus P_g = \gamma_j^2$. Если же $P_h \setminus P_g = \gamma_j^1$ и $P_g \setminus P_h = \gamma_j^2$, то дуга исходит из вершины P_h в вершину P_g . Иначе вершины P_g и P_h не связаны.

Граф достижимости удобно задать перечнем дуг и приписанных им меток переходов автомата, так как графы данного вида обладают сравнительно небольшим числом дуг. Полученный граф достижимости имеет 12 вершин и 19 дуг: $(\{1\}, \{10\}, \tau_1)$, $(\{10\}, \{2,3,4\}, \tau_2)$, $(\{2,3,4\}, \{3,4,5,6\}, \tau_3)$, $(\{2,3,4\}, \{2,3,7\}, \tau_5)$, $(\{2,3,4\}, \{2,3,9\}, \tau_6)$, $(\{3,4,5,6\}, \{4,6,8\}, \tau_4)$, $(\{3,4,5,6\}, \{3,5,6,7\}, \tau_5)$, $(\{3,4,5,6\}, \{3,5,6,9\}, \tau_6)$, $(\{2,3,7\}, \{3,5,6,7\}, \tau_3)$, $(\{2,3,7\}, \{2,3,9\}, \tau_7)$, $(\{4,6,8\}, \{6,7,8\}, \tau_5)$, $(\{4,6,8\}, \{6,8,9\}, \tau_6)$, $(\{3,5,6,7\}, \{6,7,8\}, \tau_4)$, $(\{3,5,6,7\}, \{3,5,6,9\}, \tau_7)$, $(\{2,3,9\}, \{3,5,6,9\}, \tau_3)$, $(\{6,7,8\}, \{6,8,9\}, \tau_7)$, $(\{3,5,6,9\}, \{6,8,9\}, \tau_4)$, $(\{6,8,9\}, \{11\}, \tau_8)$, $(\{11\}, \{1\}, \tau_9)$.

Более наглядно граф задается в матричном виде, где строкам и столбцам соответствуют полные состояния P_i автомата, и на пересечении i -й строки и j -го столбца указывается переход из состояния P_i в P_j . Для полученного графа эта матрица имеет следующий вид:

Таблица 1. Генерация графа достижимых состояний

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
$P_1=\{1\}$		τ_1										
$P_2=\{10\}$			τ_2									
$P_3=\{2,3,4\}$				τ_3	τ_5	τ_6						
$P_4=\{3,4,5,6\}$							τ_4	τ_5	τ_6			
$P_5=\{2,3,7\}$							τ_7	τ_3				
$P_6=\{2,3,9\}$									τ_3			
$P_7=\{4,6,8\}$										τ_5	τ_6	
$P_8=\{3,5,6,7\}$									τ_7	τ_4		
$P_9=\{3,5,6,9\}$											τ_4	
$P_{10}=\{6,7,8\}$											τ_7	
$P_{11}=\{6,8,9\}$												τ_8
$P_{12}=\{11\}$	τ_9											

Сокращение числа вершин в графе достижимости.

В настоящее время известны подходы к сокращению размерности графа достижимости (числа вершин и дуг) без потери важной информации для решаемой задачи (обзор таких методов можно найти в работе [9]). Основным недостатком предложенных правил редукции графа является то, что их применимость ограничена относительно специфическими структурами. Эффективным подходом к редукции графа достижимости является доопределение частичное упорядочение параллельных переходов (partial order reduction [14]). В основе методов сокращения графа лежит предположение о коммутативности асинхронных одновременно происходящих процессов. Такие методы применяются в методах формальной верификации сетей Петри, однако эти методы не учитывают информационное взаимодействие переходов, что неприемлемо при решении задачи генерации тестов на основе графа достижимых состояний.

Доопределение частичного порядка на множестве параллельных переходов основано на том факте, что некоторые параллельно выполняемые переходы не зависят друг от друга, в том смысле, что порядок выполнения одних переходов не влияет на условия срабатывания других. Соответственно, можно рассматривать не все возможные порядки выполнения таких переходов, а упорядочить их произвольным образом, в том числе и совместив их выполнение. Получаемый в результате граф достижимых состояний эквивалентен исходному по степени соответствия спецификации системы, но содержит значительно меньше вершин и дуг.

Пара переходов $\tau_i = (\gamma_i^1, k_i^1) \rightarrow (\gamma_i^2, k_i^2)$ и $\tau_j = (\gamma_j^1, k_j^1) \rightarrow (\gamma_j^2, k_j^2)$, одновременно срабатывающих в состоянии $S_i(W_i)$, являются совместимыми (коммутативными) в рамках этого состояния, если они могут выполняться в любом порядке, приводя в итоге к одному и тому же полному состоянию S_{i+1} автомата. Совместимость переходов τ_i и τ_j имеет место, если:

- 1) условия их срабатывания совместимы, т.е. если $k_i^1 \wedge k_j^1 \neq 0$;
- 2) изменение значений переменных в результате срабатывания каждого из этих переходов не приводит к нарушению условия срабатывания другого, т.е. в том случае, если $k_j^1 \wedge k_i^2 \neq 0$ и $k_i^1 \wedge k_j^2 \neq 0$;
- 3) изменения значений выходных сигналов, вызываемые переходами, совместимы, т.е. если $k_i^2 \wedge k_j^2 \neq 0$.

При любом порядке выполнения совместимых переходов полное состояние S_i автомата заменяется на состояние $(S_i \setminus (\gamma_i^1 \cup \gamma_j^1)) \cup (\gamma_i^2 \cup \gamma_j^2)$. Определение совместимости переходов может быть обобщено и на случай более двух переходов: переходы, возможные в некотором состоянии, могут быть выполнены одновременно, если они попарно совместимы.

Построение сокращенного графа достижимости параллельного автомата.

Если в процессе построения графа достижимости для очередной достигнутой маркировки возможно срабатывание пары (или более) параллельных переходов, не находящихся в отношении следования, то следующее полное состояние строится как результат одновременного срабатывания тех из этих переходов, которые являются совместимыми.

Предложенный в работе [13] алгоритм генерации полных состояний параллельного автомата модифицируется следующим образом. Множество полных состояний $P = \{P_1, P_2, \dots, P_l\}$ автомата строится последовательно, начиная с заданного начального состояния $P = \{P_1\}$. Также в алгоритме [13] на каждом шаге выбирается очередное уже сформированное полное состояние $P_g \subseteq P$, и для него производится последовательный просмотр всех переходов τ_i автомата. В результате

- формируется множество T_g переходов $\tau_i = (\gamma_i^1, k_i^1) \rightarrow (\gamma_i^2, k_i^2)$, достижимых из полного состояния P_g , т. е. переходов, для которых $\gamma_i^1 \subseteq P_g$;
- из множества T_g выделяются максимальные по включению подмножества $T_g^l \subseteq T_g$ совместимых переходов, образуя покрытие множества T_g этими подмножествами;
- для каждого подмножества T_g^l совместимых переходов формируется порождаемое ими полное состояние $P_{g+1}^l: P_{g+1}^l = (P_g \setminus (\cup_i \gamma_i^1 / \tau_i \in T_g^l)) \cup (\cup_i \gamma_i^2 / \tau_i \in T_g^l)$, которое объявляется новым

достижимым полным состоянием и порождает вершину графа достижимости (если оно отлично от полученных ранее);

– в граф достижимости вводится дуга из g -й вершины P_g во вновь введенную вершину P_{g+1} , порожденную подмножеством переходов из T_g^1 .

Этот процесс повторяется для каждого из вновь полученных полных состояний. Процесс заканчивается построением графа достижимых состояний, когда не получается новых полных состояний, отличных от уже полученных.

Например, из описания приведенного выше автомата видно, что в состоянии $P_3 = \{2, 3, 4\}$ возможно срабатывание трех переходов $\tau_3 = (2, -) \rightarrow (5,6, y_1)$; $\tau_5 = (4, x_1) \rightarrow (7, y_1)$ и $\tau_6 = (4, x_1) \rightarrow (9, y_2)$, соответственно $T_3 = \{\tau_3, \tau_5, \tau_6\}$. Максимальными совместимыми подмножествами переходов являются $T_3^1 = \{\tau_3, \tau_5\}$ и $T_3^2 = \{\tau_3, \tau_6\}$, они дают минимальное покрытие множества T_3 . Срабатывание переходов из $T_3^1 = \{\tau_3, \tau_5\}$ порождает новое полное состояние $P_4 = \{3, 5, 6, 7\}$, которое достижимо при срабатывании переходов τ_3 и τ_5 в любом порядке (в зависимости от сигналов, поступающих на входы автомата). Срабатывание переходов из $T_3^2 = \{\tau_3, \tau_6\}$ порождает новое полное состояние $P_5 = \{3, 5, 6, 9\}$. Соответственно, в граф достижимости вводятся вершины P_4 и P_5 и две дуги: (P_3, P_4) и (P_3, P_5) .

Полученный граф достижимости имеет восемь вершин и 10 дуг: $(\{1\}, \{10\}, \tau_1)$, $(\{10\}, \{2,3,4\}, \tau_2)$, $(\{2,3,4\}, \{3,5,6,7\}, \tau_3, \tau_5)$, $(\{2,3,4\}, \{3,5,6,9\}, \tau_3, \tau_6)$, $(\{3,5,6,7\}, \{3,5,6,9\}, \tau_7)$, $(\{3,5,6,7\}, \{6,7,8\}, \tau_4)$, $(\{3,5,6,9\}, \{6,7,8\}, \tau_4)$, $(\{6,7,8\}, \{6,8,9\}, \tau_7)$, $(\{6,8,9\}, \{11\}, \tau_8)$, $(\{11\}, \{1\}, \tau_9)$.

В матричное задание полученного графа достижимости имеет следующий вид:

Таблица 2. Генерация сокращенного графа достижимости параллельного автомата

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$P_1 = \{1\}$		τ_1						
$P_2 = \{10\}$			τ_2					
$P_3 = \{2,3,4\}$				τ_3, τ_5	τ_3, τ_6			
$P_4 = \{3,5,6,7\}$					τ_7	τ_4		
$P_5 = \{3,5,6,9\}$							τ_4	
$P_6 = \{6,7,8\}$							τ_7	
$P_7 = \{6,8,9\}$								τ_8
$P_8 = \{11\}$	τ_9							

Таким образом, в результате выполнения предложенного алгоритма удалось сократить более чем на четверть число вершин (с 12 до 8) и почти в два раза число дуг (с 19 до 10) графа достижимых состояний параллельного автомата.

Заключение.

Предложен метод построения сокращенного графа достижимых состояний параллельного автомата, который основан на предположении независимости срабатывания параллельно происходящих переходов автомата. Сокращение графа достигается доопределением частичного порядка на множестве переходов. Получаемое сокращение графа достижимости позволяет решать задачу построения тестов для сложных систем управления с параллелизмом поведения.

Список литературы

- [1]. Ebert, C. Embedded Software: Facts, Figures, and Future / C. Ebert, C. Jones // Computer. – 2009. – Vol. 42, no. 4. – P. 42–52.
- [2]. Валидация на системном уровне. Высокоуровневое моделирование и управление тестированием: пер. с англ. Е. Б. Махияновой / М. Чэнь [и др.]. – М.: Техносфера, 2014. – 296 с.
- [3]. Lee, D. Principles and methods of testing finite state machine – a survey / D. Lee, M. Yannakakis // Proceedings of the IEEE. – 1996. – Vol. 84, no. 8. – P. 1090–1123.
- [4]. Kanso, B. Compositional testing for FSM-based models / B. Kanso, O. Chebaro // Intern. J. of Software Engineering & Applications (IJSEA). – 2014. – Vol. 5, no. 3. – P. 1–20.
- [5]. Ponce de Leon, H. Model-based Testing for Concurrent Systems with Labeled Event Structures / H. Ponce de Leon, S.H. Delphine Longuet // Software Testing, Verification & Reliability. – 2014. – Vol. 24, no. 7. – P. 558–590.

- [6]. Tretmans, J. Model based testing with labelled transition systems / J. Tretmans // Formal Methods and Testing: Lecture Notes in Computer Science. – Springer, 2008. – Vol. 4949. – P. 1–38.
- [7]. Zhu, H. A Methodology of Testing High-level Petri Nets / H. Zhu, X.D. He // Information and Software Technology. – 2002. – Vol. 44. – P. 473–489.
- [8]. Liu, J. I/O Conformance Test Generation with Colored Petri Nets / J. Liu, X. Ye1, J. Zhou, X. Song // Applied Mathematics and Information Sciences. – 2014. – Vol. 8, no. 6. – P. 2695–2704.
- [9]. Karatkevich, A. Dynamic Analysis of Petri Net-based Discrete Systems / A. Karatkevich. – Berlin : Springer-Verlag, 2007. – Vol. 358. – 166 p.
- [10]. Закревский, А. Д. Параллельные алгоритмы логического управления / А. Д. Закревский. – Минск: Ин-т техн. кибернетики НАН Беларуси, 1999. – 202 с.
- [11]. Питерсон, Дж. Теория сетей Петри и моделирование систем: пер. с англ. М. В. Горбатовой, В. Л. Торхова, В. Н. Четверикова / Дж. Питерсон. – М.: Мир, 1984. – 264 с.
- [12]. Hack, M. Analysis of production schemata by Petri nets / M. Hack // Project MAK-94. – Cambridge, 1972. – 119 p.
- [13]. Поттосин, Ю. В. Верификация систем с параллелизмом поведения на основе графа достижимых состояний / Ю.В. Поттосин, В.И. Романов, Л.Д. Черемисинова // Информатика. – 2019. – Т. 16, № 2. – С. 62–72.
- [14]. Lluch-Lafuente, A. Partial Order Reduction in Directed Model Checking / A. Lluch-Lafuente, S. Edelkamp, S. Leue // Proceedings of the 9th International SPIN Workshop on Model Checking of Software. Berlin, Heidelberg: Springer-Verlag, 2002 April. – Vol. 11–13. – P. 112–127.

MINIMIZATION OF THE REACHABILITY GRAPH FOR A PARALLEL AUTOMATON

L.D. Cheremisinova

*Principal researcher of UIIP of NAS of Belarus, doctor
of technical sciences, professor*

*United Institute of Informatics Problems of National Academy of Sciences of Belarus,
Republic of Belarus
E-mail: cher, cld}@newman.bas-net.by*

Abstract. The problem of testing control systems, which are inherent in the parallelism of the processes occurring in them, is considered. The parallel automaton model, which is a subclass of colored Petri nets, is used as a language for specifying such systems. The analysis of the behavioral properties of a parallel automaton is carried out on the basis of its reachability graph, the vertices of which correspond to all possible states markings, the number of which grows exponentially with an increase in the degree of parallelism of states. The paper proposes a method for minimizing the reachability graph by reducing the number of its vertices based on the introduction of the extension of the partial order on the set of parallel firing the automaton transitions.

Keywords: parallel automaton, reachability graph, discrete device testing.