

УДК 519.179.2, 004.94, 004.451.24

¹В.В. Кузьмук, ²А.В.Кузьмук, ¹О.А. Супруненко, ³Е.А. Тараненко¹Черкасский национальный университет имени Богдана Хмельницкого, Черкасы²Институт проблем моделирования в энергетике АН Украины, Клев³ООО «Алтимед»

МОДИФИЦИРОВАННЫЕ СЕТИ ПЕТРИ И СОВРЕМЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ В СЛОЖНЫХ СИСТЕМАХ

Рассматривается моделирование параллельных процессов в вычислительных системах с помощью аппарата сетей Петри. Предлагается проводить построение модели параллельных процессов на основе интерпретаций сетей Петри. Приведены примеры моделей.

Ключевые слова: параллельные процессы, модель параллельной системы, интерпретации сети Петри

Введение

Активное развитие многопроцессорных и многомашинных систем, в том числе многоядерных и кластерных вычислительных систем, требует создания соответствующего алгоритмического аппарата и программных средств для построения, верификации и тестирования параллельных программ [1].

Использование удаленного доступа к многопроцессорным суперкомпьютерам требует использования параллельных программ, которые адаптируются к произвольной архитектуре компьютерной системы [2;3]. Особенно следует выделить ряд задач: исследование структуры программы, локализация кода для ввода-вывода, выделение участков потенциального параллелизма, определение и минимизация точек перераспределения данных, разбиение общего массива вычислений на локальные, обрабатываемые разными вычислительными элементами.

Разработка параллельных программ на сегодняшний день является очень трудоёмким процессом. Для его облегчения создаются библиотеки эффективных процедур и алгоритмов под конкретные архитектуры вычислительных машин [2]. Множество проблем возникает также при тестировании параллельных программ как на машинах, на которых создавался программный код, так и на конечных суперкомпьютерах, для которых он создавался.

Для эффективного построения, верификации и тестирования параллельных программ в современном программном обеспечении активно используется модельный подход [4], который предполагает создание и тестирование структуры параллельной программы на ее модели, которая

построена с использованием соответствующих графоаналитических средств моделирования. В качестве основы для моделирования параллельных алгоритмов в данной работе предлагается использовать Сети Петри (Petri Netzen), созданные Карлом Адамом Петри и предназначенные для моделирования асинхронных параллельных процессов [5].

Моделирование параллельных процессов сетями Петри

При определении понятия параллельных процессов исходным является событие [1]. Совокупность упорядоченных по времени событий называется последовательностью событий. Каждое событие включает два взаимно связанных частных события: «начало» и «конец» (рис. 1, а). Последовательность событий, которая имеет начало и конец, а также может быть прервана во времени, называется *действием* (a_i).

Совокупность упорядоченных во времени действий называется *последовательностью* действий (ПД), управление которой может осуществляться событиями от внешнего процесса или временными событиями. Два действия называются *последовательными*, если событие «начало» последующего действия зависит и наступает позднее (или одновременно с окончанием) события «конец» рассматриваемого в данный момент действия (a_1 и a_3 , a_3 и a_4). В данном случае говорят о причинностной связи действий, которые выполняются последовательно одно за другим.

Два действия называются *параллельными*, если они не имеют причинностных связей и могут выполняться одновременно (a_1 и a_2 , a_2 и a_4), т. е. в момент *начала* или *выполнения* одного действия начинается выполнение второго.

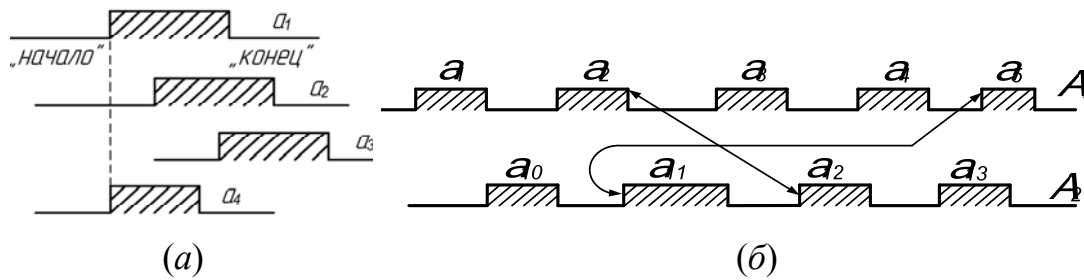


Рис. 1 Последовательные и параллельные события (а), частично-зависимые параллельные последовательности (б)

Во многих задачах, связанных с функционированием нескольких последовательностей действий, имеют место ситуации, когда некоторые действия одной последовательности в определённые моменты времени зависят от начала или окончания действий в другой последовательности действий, например последовательности действий \bar{A}_1 и \bar{A}_2 на рис. 1, б. Такие последовательности действий называются *частично-зависимыми параллельными последовательностями*.

Подклассом частично-зависимых параллельных процессов является *подчиненные последовательности действий*. Если зависимости между отдельными действиями не меняются во времени, то они образуют параллельно-последовательные последовательности действий. В большинстве работ под *параллельными* понимают частично-зависимые параллельные и параллельно-последовательные последовательности действий. В случае, когда последовательности действий не имеют между собой причинностных связей, их называют *независимыми*. Если при этом они могут быть смещены во времени без ущерба для поставленной задачи и разбиты на целое число выбранных временных интервалов тактовой частоты, говорят о *квазипараллельной последовательности действий* (рис. 2).

На рис. 2 действие «вычисления X или Y» обозначаются штриховкой //, а действие «выдача результата» обозначается \\\.

Совокупность конечных последовательностей действий, предназначенных для задач управления, называется *процессом управления (ПрУ)*. Параллельные процессы управления отличаются от параллельных процессов вычисления тем, что в системах управления дополнительно на процесс вычисления влияют сигналы управления из внешней среды $X = \{x_1, x_2, \dots, x_b, \dots, x_l\}$ по аналогии с множеством входных сигналов конечных автоматов X_i .

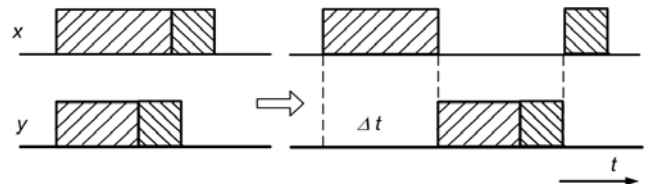


Рис. 2 Квазипараллельная последовательность действий

Вышеизложенное позволяет сформулировать пять классов параллельных процессов (ПП).

1. Синхронные ПП – последовательности действий (ПД), в которых продолжительность их параллельного выполнения определена во времени или если действие в одном процессе зависит от состояния других взаимодействующих с ним ПП.
2. Асинхронные ПП – последовательности действий, в которых продолжительность параллельного их выполнения не зависит от ПД в других ПП.
3. Независимые ПП – последовательности действий, которые не зависят от выполняемых действий в других ПП.
4. Подчиненные ПП – последовательности действий, которые управляются действиями от других ПП более высокого уровня.
5. Коллективные ПП – последовательности действий, представленные древовидными графами и соответствующие выполнению общей задачи несколькими группами «коллектива вычислителей». Действие руководителя заканчивается, когда выполнены и состыкованы все распараллеленные ПД.

В работе [1] в качестве *параллельных процессов* рассматривались *асинхронные параллельные процессы*. Особый интерес представляют реальные параллельные процессы управления, объединенные различными причинностными связями в одну общую систему, которые в основном принадлежат к классу асинхронных ПП.

Одним из графоаналитических методов описания и моделирования параллельных процессов и взаимодействий между ними является теория сетей Петри (Petri - Netze, PN) [5]. Важным

моментом в теории *PN* является то, что причинностные связи, включающие «причину – действие – обозначение», представляют интерес при моделировании как параллельных, так и параллельно-последовательных процессов, позволяют отображать их уникальные особенности, чем вносят разнообразие в варианты решения задачи. Особенностью *сетей Петри* или *графов Петри* является наличие двух видов несовместных вершин: вершин переходов t_i и вершин мест (узлов) p_e , соединенных между собой ориентированными дугами (ребрами) по определенным правилам S . В графах Петри *переходам* t_i соответствовали одноименные *автоматы* A_i , а узлам p_e – *условия активизации* или *после активизации* (p_f) автоматов A_i .

В период активного распространения графов Петри не уделялось должного внимания формализации, однозначного соответствия элементов графов соответствующим действиям, условиям их «начала», и последующим условиям «конца», правилам построения графов как ПА и правилам их обхода, рекомендациям их программной или аппаратной реализации. К.А. Петри предложил изображать вершины переходов графически соответствующими *квадратами*, а вершины мест или узлы p_e –

кружками (рис. 3,*а*). В дальнейшем из-за большей наглядности квадраты были заменены на перпендикулярные к дугам утолщенные линии – сплюснутые квадраты (рис. 3,*б*). Наличие меток в соответствующих вершинах p_e или t_i позволяет визуально наблюдать в каких точках параллельных процессов находится вычислительная система при моделировании их с помощью *сетей Петри*, т.е. отслеживать динамику выполнения (отработку) параллельных процессов. При этом один раз установленные последовательности не всегда являются оптимальным решением задачи. Изменение решения возможно, когда известно, какие элементы выбранных последовательностей могут быть переставлены и выполнение каких условий должно быть обеспечено. Для удобства построения и изменения параллельно-последовательных моделей процессов К.А. Петри сформулировал правила взаимодействия автоматов. Каждый абстрактный автомат A_i в такой системе, обрабатывая заданную смену состояний и взаимодействуя с другими автоматами A_j , позволяет построить модель параллельно-последовательного процесса. Основным смыслом теории К.А. Петри состоит в формулировке так называемых систем «условия – действия» (ВЕ-систем).

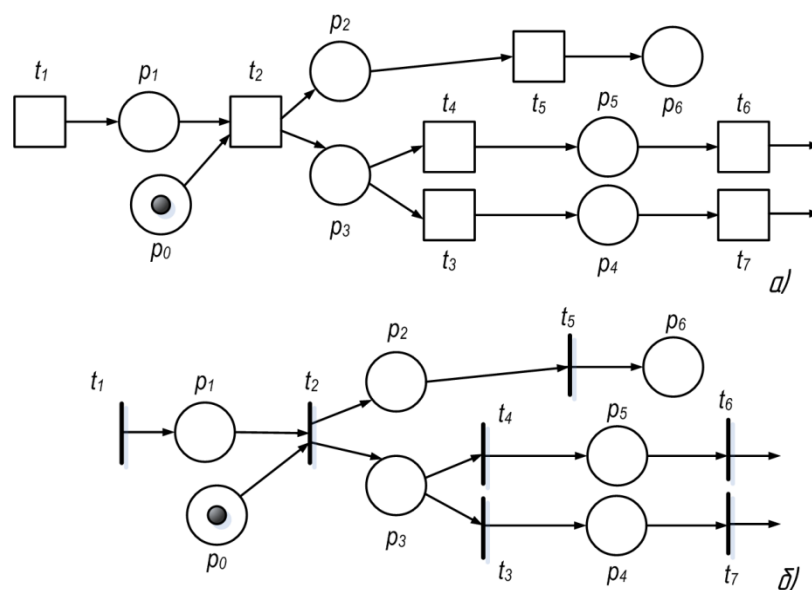


Рис. 3. Сеть Петри на основе элементов, предложенных К.А. Петри (а) и в современной нотации (б)

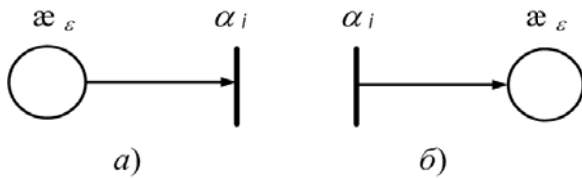


Рис. 4 ВЕ-системы, которые характеризуются парами «условие – действие» (а) и «действие – условие» (б).

ВЕ-системы характеризуются парами «условие – действие» (рис. 4, а) и «действие – условие» (рис. 4, б), которые алгоритмически описывают протекающие процессы и характеризуются наличием меток в соответствующих вершинах мест p_f .

Сети Петри (PN) являются специализированными бихроматическими (двудольными) графами, позволяющими моделировать и алгоритмически описывать динамику протекания параллельных процессов и причинностные связи между ними. В настоящее время известны оригинальные и разрабатываются новые интерпретации и модификации сетей Петри для различных областей применения. Под *интерпретацией* сетей Петри понимают определенный набор понятий и правил, в основе которых лежит принцип построения размеченных графов, состоящих из двух типов вершин (множества вершин переходов T и множества вершин мест P) соединенных между собой дугами K по определенным функциональным правилам S . В общем случае сети Петри $PN = (P, T, K, S)$, как и любые графы, содержат структурные (P, T, K) и функциональные (S) элементы.

Двумя основными задачами в работах [1; 7; 8] были рассмотрены задача определения методов сравнения различных интерпретаций сетей Петри и задача выбора наиболее подходящих PN для моделирования мультипроцессорных систем управления и алгоритмического моделирования, протекающих в них параллельных процессов. Сегодня говорить о сетях Петри вообще означает то же, что и говорить, в общем, об автоматах. Известно, что автоматы бывают детерминированные и недетерминированные, конечные и бесконечные, синхронные и асинхронные, вероятные, самонастраивающиеся, обратимые, автономные, автоматы Мили и Мура и т. д. [1]. Широкое распространение сетей Петри привело к тому, что многие исследователи начали их использовать для решения тех или иных задач. Благодаря этому были сформулированы ВЕ-сети (EN), Макро-Е-сети (MEN), О-сети (O - Net), Про-сети (Pro - Net), цветные сети (CPN), временные сети (TP), биполярные сети (BP), автоматные сети (SM), MG-сети, свободно-выбранные сети (FC), сильно связанные сети (SQ), оценочные сети (BPN), управляющие сети (SN), сети контроля

протекания (AKN), регулярные, иерархические и многие другие [9].

В работах [1, 7-9] была предпринята попытка найти общие структурные и функциональные элементы в известных интерпретациях сетей Петри для формализации их исследования. Остановимся подробнее на основных составляющих, которые лежат в основе правил построения и отработки той или иной Сети Петри.

Структурные элементы включают множество вершин мест $P = \{p_e\}$ и переходов $T = \{t_j\}$, а также множество дуг $K = \{k_q\}$, соединяющих эти вершины. Дуги k в PN соединяют лишь разноименные вершины, т. е. $P \cap T = T \cap P = \emptyset$. Это вытекает из посылки, что в ВЕ-системах из одного условия нет перехода к условию, а есть переход к действию и наоборот (рис. 5).

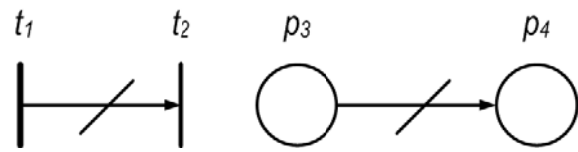


Рис. 5 Варианты переходов, запрещённые в ВЕ-системах

Функциональные правила $S = \{w, b, m, L\}$ включают элементы, на которых базируются правила построения и реализации той или иной интерпретации PN. *Оценочная* или *весовая* функция $w : K \rightarrow N$, где N – множество натуральных чисел, указывает на кратность дуг $k_q \in K$ и определяет число меток, которые одновременно по ним протекают. Весовая функция w показывает, сколько раз должно повториться срабатывание перехода t_j или сколько меток должно содержать место p_e , чтобы сработал последующий переход t_{j+1} . Если вес дуги $w = 0$, то срабатывание перехода t_j может наступить в случае, если разметка места, из которого выходит эта дуга, равна $m(p_e) = 0$. При этом в сети Петри имеется *запрещающая* (inhibit) или *инверсная* дуга (k_2), которая оканчивается не стрелкой, а крупной точкой (рис. 6).

Функция обозначения $b : T \rightarrow L$ позволяет связать срабатывание переходов $t_j \in T$ с опросом входных i_x и выдачей выходных i_y сигналов управления.

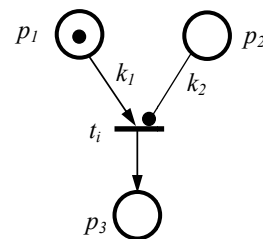


Рис.6 Переход с инверсной дугой

Функция обозначения указывает дальнейший путь, по которому начнут перемещаться метки, и действия a_{j+q} которые последуют. Она позволяет установить приоритет для срабатывания переходов t_j . Разметочная функция $m: P \rightarrow N \cup \{0\}$ ставит в соответствие каждому месту $p_e \in P$ целое число меток N или 0 и определяет его текущую разметку $m(p_e) = n_v \in N \cup \{0\}$. Множество $L = X \cup Y$ определяет входной $X = \{x_1, x_2, \dots, x_s\}$ и выходной $Y = \{y_1, y_2, \dots, y_v\}$ алфавиты сигналов управления.

Формулировка структурных и функциональных компонент сетей Петри позволила сформулировать [1; 9] *Общие (классические) сети Петри PN*, содержащие указанные элементы. Их особенностью является то, что они с помощью модификации того или иного элемента позволяют перейти к большинству известных интерпретаций. В Общих сетях Петри каждый переход t_j моделирует действие a , вершина p_e , связанная направленной дугой с t_j , моделирует одно из условий E_e , при котором может начаться выполнение действия a , а одна метка в p_e указывает на наличие одного частичного условия $iE_e \in E$.

Анализ средства решения задачи

Сравнение составляющих частей сети Петри и характеризующих параметров алгоритма позволило сформулировать понятие параллельного алгоритма [1]. *Параллельным алгоритмом* (ПА) называется система формальных правил, характеризующаяся следующими независимыми параметрами: 1) совокупность возможных исходных данных; 2) совокупность возможных результатов обработки; 3) совокупность возможных промежуточных результатов и возможных их параллельных последовательностей; 4) правило начала обработки; 5) правило непосредственной обработки; 6) правило окончания обработки; 7) правило извлечения текущих и конечных результатов; 8) правило взаимодействия частичных алгоритмов в параллельном алгоритме; 9) правило взаимодействия параллельного алгоритма с другими алгоритмами и параллельными алгоритмами.

Введение двух дополнительных параметров 8 и 9, а также уточнение 3-го параметра для объяснения понятия параллельного алгоритма несколько сузили классическое понятие последовательного алгоритма, но в то же время расширили область применения теории алгоритмов и позволили применять ее для представления динамики параллельных процессов. Отличие между двумя последними правилами заключается в том, что взаимодействия между параллельно (одновременно) обрабатываемыми частичными алгоритмами в

параллельном алгоритме характеризуются алгоритмическими связями, а взаимодействия параллельных алгоритмов с другими алгоритмами и параллельными алгоритмами могут быть реализованы как с помощью алгоритмических, так и с помощью информационных связей [1; 9].

Рассмотрим связь между структурными и функциональными элементами PN и характеризующими независимыми параметрами параллельных алгоритмов. Используемый алфавит L включает входной алфавит сигналов управления X , который отображает совокупность возможных исходных данных в ПА, и выходной алфавит сигналов управления Y , который характеризует совокупность возможных результатов обработки ПА. Разметочная функция m характеризует текущее общее состояние параллельного алгоритма и соответствует совокупности возможных промежуточных результатов, а множество возможных раз меток $M = \{\vec{m}_x\}$ соответствует совокупности возможных их параллельных последовательностей A_1, \dots, A_j в ПА. Наличие меток в p_e определяет текущие состояния, в которых находится мультипроцессорная система. Для отличия понятия состояния, характерного для теории автоматов, сети Петри характеризуются *случаем Z* – совокупностью законченных действий и действий, которые еще выполняются.

Случаем Z в сетях Петри называют совокупность всех установившихся и неуставившихся состояний $Z = \{Z_1, Z_2, \dots, Z_n\}$, в которых находится в данный момент мультипроцессорная система.

К установившимся относятся состояния, характеризующиеся фиксацией метки или группы меток в месте p_e в данный момент времени, а к неуставившимся — состояния, когда метки из одного или нескольких мест p_e изъяты, а в последующие места p_q еще не перенесены, т. е. переход системы из состояния Z_e в состояние Z_p еще не закончился (не закончились действия a_i в ПА). Метки в PN служат для обозначения выполнения каких-то действий a_i , выполнения условий $i+1E$ для последующего срабатывания переходов t_{i+n} – выполнения последующих действий и служат механизмом синхронизации в тех ветвях параллельного алгоритма, по которым они проходят. Правила начала обработки ПА определяются в PN функций $\phi(w, m, X)$. Функция обозначения b реализует правила непосредственной обработки отдельных действий a_i в ПА. Правила окончания обработки ПА определяются в PN функцией $\psi(m, b, Y)$. Правило взаимодействия частичных алгоритмов A_i и A_v в ПА реализуется в

сетях Петри с помощью введения дополнительных дуг k , соединяющих в зависимости от решаемой задачи разноименные вершины в A_i и A_j . Правило взаимодействия ПА с другими A_i или ПА' может быть реализовано как с помощью дополнительных дуг k , определяющих алгоритмические условия, так и с помощью дополнительных воздействий x (y), задающих дополнительные условия и реализующих информационные связи.

Таким образом, общие (классические) сети Петри можно рассматривать не только как математический аппарат для построения аналитических, имитационных алгоритмических моделей, но и как алгоритмический метод представления и моделирования параллельных процессов в мультипроцессорных системах контроля, измерения и управления, работающих в реальном масштабе времени.

Основными составляющими параллельных алгоритмов, как MIMD-систем, являются поток команд и поток данных. Аппарат классических сетей Петри хорошо приспособлен для управления потоком команд, но требует адаптации для эффективного управления потоком данных. Модификация классических сетей Петри направлена на решение этой задачи, а набор статических и динамических свойств обеспечивает качественное тестирование созданных моделей параллельных программ. Статические свойства сетей Петри позволяют отследить ошибки в логике построения параллельной программы и некоторые виды тупиковых ситуаций [1]. Динамические свойства сетей Петри, которые проявляются в режиме имитации модели параллельного алгоритма, служат для выявления неживых веток, deadlock-ов, проверки безопасности вершин мест, выявления конфликтных ситуаций. Это особенно важно при наличии в алгоритме скрытых конфликтов, когда их невозможно выявить в процедуре стандартного тестирования готового программного продукта. Они устраняются при выявлении ситуации клиентом на этапе эксплуатации программного продукта.

Пример моделирования управляющей системы

Рассмотрим на примере возможное взаимодействие нескольких асинхронных параллельных процессов (ПП) в одной автоматизированной системе. Очень удачным примером (с точки зрения наглядности и простоты) является описание модели функционирования кекс-автомата [10], представленной в работе доктора Райсига. В данной модели (рис. 7) отображаются процессы, происходящие в кекс-автомате при осуществлении и выдаче заказа – покупке кексов. На начальном этапе происходит приём заказа и денег (вершина перехода ϵ). Параллельные процессы при обработке

заказа осуществляются от вершины a по двум направлениям: подсчёт денег для выдачи сдачи и заказ необходимого количества кексов. Эти процессы взаимозависимы, поскольку сдача (или возврат денег) зависит от количества кексов в накопителе – ответ получаем после активизации вершины перехода b и обрабатываем через вершину места «Нет сигнала» в вершине перехода a . Данный пример демонстрирует простоту и наглядность представления и анализа модели при её отражении сетями Петри.

Рассмотрим модель кофе-автомата, построенную с использованием сетей Петри. Предположим, Вы захотели выпить стаканчик горячего кофе или чая. Вы опускаете деньги в купюро- или монетоприемник. Автомат принимает деньги и их проверяет. Если принятые деньги не фальшивые и их сумма может оплатить Ваш заказ.

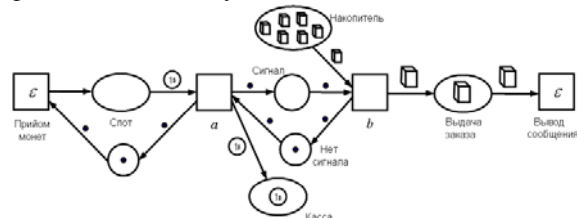


Рис. 7. Модель кекс-автомата, представленная сетью Петри

Вы можете выбрать в рамках этой суммы желаемое кофе и нажать соответствующую кнопку выбора. В место для налива кофе опустится чистый стаканчик и в него начнет наливаться кофе. После того, как стаканчик наполнится кофе, загорается зеленая лампочка, которая сигнализирует о том, что можно вынимать кофе из кофейного автомата и его пить. Во время, когда Вы пьете кофе, Вы можете захотеть выпить еще один стаканчик, и чтобы не терять время можете заказать его, не прекращая пить первый стаканчик, - оплачивать и наливать. Вот здесь и появляется второй параллельный процесс – процесс наполнения второго стаканчика кофе во время потребления первого стаканчика (рис. 8), который хоть и параллельный, но следует четко заложенному в кофе-автомате алгоритму. Алгоритм включает четкие последовательности действий (оплата, выбор понравившегося кофе, опускание чистого стаканчика, наливание выбранного кофе в стаканчик, выемка наполненного кофеем стаканчика, и.т.д.) Если эта последовательность действий не соблюдается: 1) Вы вынимаете стаканчик во время наливания кофе (не загорелась зеленая лампочка разрешения выемки наполненного кофе стаканчика) - кофе выльется в канализацию, а стаканчик останется пустой или наполовину пустой.

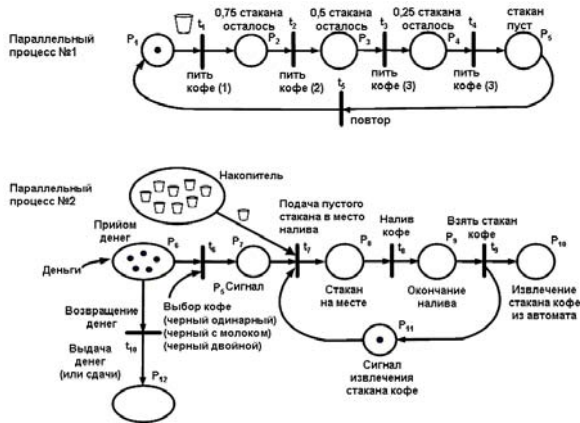


Рис. 8 Модель функционирования кофейного автомата

2) Вы не опустили деньги, но хотите получить кофе – эту последовательность действий автомат не выполнит.

Таким образом, два параллельных процесса, хотя и параллельны, и могут выполняться одновременно, существуют правила их взаимодействия, невыполнение которых может привести к отрицательным результатам. Существуют реальные «причинностные связи» [1; 9], которые связывают два параллельных процесса.

Выводы

Многообразие интерпретаций и модификаций сетей Петри красноречиво говорит об их большом потенциале в области моделирования параллельных процессов в сложных системах. Основными задачами на сегодняшний день являются создание средств для адаптации графоаналитического аппарата сетей Петри к решению прикладных задач. К данным задачам относятся: адаптация интерпретаций и модификаций сетей Петри к эффективному моделированию данных и их преобразование в параллельных вычислительных системах, оптимизация структуры параллельных программ на PN-моделях, поиск необходимого набора тестов на модели для верификации параллельных программ.

Названные задачи охватывают область аппаратного и программного обеспечения автоматизированных систем. Примерами таких задач являются аппаратно-программные системы контроля доступа в компьютерных сетях; современные программные системы специализированного назначения, например медицина [11], геоинформатика, нанотехнологии, предназначенные для работы на многопроцессорных системах обработки информации; программные и аппаратно-программные системы защиты информации [12] и многие другие.

Список литературы

1. Кузьмук В.В., Супруненко О.О. Модифицированные сети Петри и устройства моделирования параллельных процессов: Монография. – К.: Маклаут, 2010. – 260 с.
2. Карпов А. Введение в проблематику разработки параллельных программ. [Электронный документ]. Адрес: <http://www.viva64.com/ru/a/0016/>. Проверен: 29.12.10.
3. Антонов А.С., Воеводин В.В. Эффективная адаптация последовательных программ для современных векторно-конвейерных и массивно-параллельных супер-ЭВМ. // Программирование. 1996, № 4, с. 37-51.
4. Карпов Ю.Г. Model Checking. Верификация параллельных и распределённых программных систем. – СПб.: БХВ-Петербург, 2010. – 560 с.
5. Petri C.A. Kommunikatoin mit Automaten. – Bonn: Institut für Instrumentelle Mathematik, 1962. – 89 S.
6. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация, применение. – СПб: БВХ - Петербург, 2003. – 1104 с.
7. Kuzmuk V.V. Representation and Modeling of Parallel Control Processes by Means of Petri Nets. "Gordon and Breach" SP in United Kingd. Electronic Modeling, V.4(5), 1985, p.1093-1108.
8. Kuzmuk V.V. Concurrent Algorithm construction and Optimization using Control Nets. Proc. 11-th IMACS World Congress on System simulation and scientific computation, 5-9 August, V3, 1985, Oslo, Norway, 1985, V.3, p.179-181.
9. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем – К: Наукова думка, 1990 – 216 с.
10. W. Reisig. Petrinetze. Vieweg+Teubner Verlag, Wiesbaden GmbH, 2010,- 247S.
11. Билякова Н.В., Кузьмук В.В., Супруненко О.А. Графоаналитическая модель биологического процесса на основе функционально-дифференциальных методов. // Матеріали XI конференції з біоніки, біокібернетики та прикладної біофізики. – К.: Вид. НТУУ «КПІ», 2010. – С. 12.
12. Супруненко О.О. Модифікація підсистем захисту інформації на основі мереж Петрі. // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Тематичний випуск: Нові рішення в сучасних технологіях. – Харків: НТУ «ХПІ», 2010. - № 57. – С. 173-177.

Статья поступила в редколлегию 23.02.2011

Рецензент: д-р техн. наук, проф. Ю.Н. Тесля, Киевский национальный университет строительства и архитектуры, Киев