немаловажную роль в оценке функционала земельных работ играет точная константа K неравенства для норм производных функций.

ЛИТЕРАТУРА

1. *С. М. Никольский* Приближение функций многих переменных и теоремы вложения.- Москва: Наука, 1977.- с.140-161, 227-232.
2. *Л. С. Понтрягин*. Математическая теория оптимальных процессов.- Москва: Наука.- 1976.- с.5-9, 221-240.
3. *Н. П. Корнейчук*. Экстремальные задачи теории приближения.- Москва: Наука, 1976.- с. 28, 34, 94-104, 176-187.
4. *В.Ф. Бабенко, Н.П. Корнейчук, В.А. Кофанов, С.А. Пичугов*. Неравенства для производных и их приложения. – Киев, Наукова думка, 2003.

пост. 24.05.07.

Сетевое описание сложных дискретных систем

Б.Б. НЕСТЕРЕНКО, М.А. НОВОТАРСКИЙ

Институт математики НАН Украины

Работа посвящена описанию структуры и принципов функционирования новой версии сетевого формального описания, ориентированного на построение имитационных моделей сложных дискретных систем, позволяющих использовать реальную рабочую нагрузку. Рассмотрены принципы определения эквивалентности сетей и предложен алгоритм определения строгого взаимного подобия сетевых описаний.

Робота присвячена опису структури і принципів функціонування нової версії мережного формального опису, орієнтованого на побудову імітаційних моделей складних дискретних систем, що дозволяють використовувати реальне робоче навантаження. Розглянуто принципи визначення еквівалентності мереж і запропонований алгоритм визначення строгої взаємної подібності мережних описів.

This paper is devoted to the description of structure and principles of functioning for the new version of the network formal description focused on simulations of complex discrete systems, allowing to use real working loading. Principles of definition of equivalence for networks are considered and the algorithm of definition of strong bisimulations for network descriptions is offered.

1. Введение

В результате развития современных технологий в наш мир приходят все новые и новые объекты, которые, в большинстве случаев, следует отнести к сложным дискретным системам. Область применения сложных дискретных систем также стремительно расширяется. Сейчас она включает весь диапазон высокотехнологичных изделий, начиная от бытовых цифровых фотоаппаратов или плееров и заканчивая самолетами или космическими кораблями. Общее для всех этих систем состоит в том, что их можно представить в виде множества состояний, эволюционирующих путем переходов от состояния к состоянию под воздействием событий. Это базовое свойство дискретных систем легло в основу практически всех формальных средств их описания. Хорошо исследовано представление дискретных систем с помощью цифровых конечных автоматов [1], кусочно-линейных агрегатов [2] и других средств формального описания, в основе графической интерпретации которых лежит теория графов. Независимо от несколько различных принципов формирования очередного состояния, общим для упомянутых формальных средств является подход, задающий функционирование дискретной системы в виде единого процесса. Однако современные сложные дискретные системы состоят из параллельно функционирующих и асинхронно взаимодействующих компонент. Поэтому представление их в виде единого процесса вызывает существенные затруднения, поскольку ведет к взрывоподобному увеличению количества его состояний. Естественным решением данной проблемы стали предложенные в 60-х годах прошлого столетия сети Петри [3]. Благодаря введению новой графической интерпретации, базирующейся на двудольных графах, они дали возможность описывать поведение дискретных систем с параллельно функционирующими компонентами. Однако вскоре стали очевидными также и существенные недостатки сетей Петри, проявляющиеся в громоздкости описания, наличии тупиковых ситуаций и ограниченном наборе правил срабатывания переходов. Стремление преодолеть указанные недостатки стало движущей силой к развитию новых формальных средств описания сложных дискретных систем. Дальнейшие исследования развивались как по пути эволюционных изменений сетей Петри, так и по пути создания принципиально новых видов сетей. Одним из удачных решений, расширяющим возможности формальных средств описания параллельных процессов и систем, стали E-сети [4]. Принципиальное отличие E-сетей состоит в том, что их синтаксис расширен за счет введения специализированных переходов, позволивших улучшить семантические правила сетевого формального описания. В E-сетях впервые также была введена процедура обработки, регламентирующая порядок обработки данных в переходе. Однако, в связи со специализацией переходов, роль таких процедур была весьма ограничена. Упомянутые ограничения стали одной из причин дальнейшего развития параллельных средств формального описания, получивших название PRO-сетей [5]. Суть данного подхода состоит в отказе от задания жестких правил функционирования переходов за счет введения специальных процедур, управляющих режимами работы переходов и обработкой информации.

Асинхронные PRO-сети или APRO-сети являются новым шагом в развитии формальных сетевых описаний. Отличие APRO-сетей состоит в детальном описании структур данных, соотнесенных с объектами сети, и процедур обработки этих данных. Такой подход позволяет строить имитационные модели, адекватно отображающие вопросы взаимодействия параллельных компонент и использующие реальную рабочую нагрузку.

2. APRO-сети

Представим APRO-сеть в виде кортежа:

$$\Phi = (P, T, F, M, V), \tag{1}$$

где $P = \{p_i\}_{i=1}^{n}$ – конечное множество позиций, $T = \{t_j\}_{j=1}^{m}$ – конечное множество переходов, $F = (P \times T) \cup (T \times P)$ – множество ребер между переходами и позициями, $M = \left\{ \left(p_k, \{\mu_l\}_{l=1}^{Max\_p_k} \right) \right\}_{k=1}^{n}$ – конечное множество маркирований, $V = (\Delta, \Psi, \Lambda)$ – множество глобальных переменных.

Графические элементы APRO-сети представлены в таблице 1. Подобно графическим обозначениям сетей Петри, позиция APRO-сетей обозначается кружком, простой переход – линией, связи – линиями со стрелками, а метка – точкой. Дополнительно введены обозначения, связанные с элементами операторного перехода. Сам операторный переход изображается в виде прямоугольника, а входы и выходы операторного перехода представлены в виде правого и левого полукругов.

Таблица 1

Обозначение			Графическое изображение	Название
$p \in P$				Позиция
$t \in T$	τ			Простой переход
	θ			Операторный переход
	θ	e_θ		Вход операторного перехода
		x_θ		Выход операторного перехода
$(p,t);(t,p) \in F$			→	Ребро
$\mu \in M$			•	Метка

Позиции APRO-сети: $p_i = \{d_i, q_i\}$, где $d_i = \{Id\_p_i, \varphi\_p_i, Cur\_p_i, Max\_p_i, \delta\_p_i\}$ – множество параметров позиции, содержащее такие элементы: Id_p_i – идентификатор позиции, φ_p_i – множество допустимых типов меток, Cur_p_i – текущее количество меток на позиции, Max_p_i – максимально допустимое количество меток, δ_p_i – локальный счетчик времени позиции; q_i – множество меток, размещенных на данной позиции.

Переходы APRO-сети включают два класса переходов: $t = \{\tau, \theta\}$, где τ – класс простых переходов, θ – класс операторных переходов. Простой переход описывает множество элементов: $\tau_j = \{\chi_j, N_j\}$, где $\chi_j = \{Id\_\tau_j, \delta\_\tau_j, \Delta\_\tau_j, Level\_\tau_j\}$ – множество параметров перехода, состоящее из таких элементов: Id_τ_j – идентификатор перехода, δ_τ_j – локальный счетчик времени перехода, Δ_τ_j – период простоя, $Level\_\tau_j$ – уровень перехода.

Функциональное ядро перехода: $N_j = \{\rho_j, \pi_j, \gamma_j, \omega_j, A_j, O_j\}$, где ρ_j – процедура активации перехода, π_j – процедура обслуживания перехода, γ_j – процедура деактивации перехода, ω_j – процедура ожидания, A_j – частично упорядоченная последовательность активностей, O_j – частично упорядоченная последовательность выходных меток.

Класс переходов θ обеспечивает композиционные свойства сети и представляет собой множество:

$$\theta = (P_\theta, T_\theta, E_\theta, X_\theta, F_\theta), \tag{2}$$

где $P_\theta = \{(p_\theta)_1, \cdots (p_\theta)_a\}$, $a \in N$ – множество позиций; $T_\theta = \{(\tau_\theta)_1, \cdots (\tau_\theta)_b\}$, $b \in N$, $T_\theta \neq \varnothing$ – непустое множество переходов; $E_\theta = \{(e_\theta)_1, \cdots (e_\theta)_c\}$, $c \in N$, $E_\theta \neq \varnothing$ – непустое множество входов;
$X_\theta = \{(x_\theta)_1, \cdots (x_\theta)_d\}$, $d \in N$, $X_\theta \neq \varnothing$ – непустое множество выходов;
$F_\theta = (P_\theta \times T_\theta) \cup (T_\theta \times P_\theta) \cup (E_\theta \times T_\theta) \cup (T_\theta \times X_\theta)$ – множество ребер.

Таким образом, операторный переход θ содержит APRO-сеть Φ_θ, определяемую как сеть нижнего уровня по отношению к сети Φ. Переходы сети Φ_θ также могут быть операторными переходами, что позволяет строить иерархические модели с неограниченным количеством уровней.

Во избежание путаницы элементы операторного перехода будем отмечать индексом с его наименованием. Формальное описание произвольной внутренней позиции $(p_\theta)_i \in \Phi_\theta$ операторного перехода θ соответствует формальному описанию произвольной позиции верхнего уровня $p_i \in \Phi$. Внутренние переходы $(\tau_\theta)_j \in \Phi_\theta$ также имеют идентичное формальное описание с простыми переходами верхнего уровня $\tau_j \in \Phi$.

Входы операторного перехода: $(e_\theta)_j = \left\{(\eta_\theta)_j, (N_\theta^e)_j, \mu_\theta\right\}$, где $(\eta_\theta)_j = \left\{(Id\_e_\theta)_j, (\delta\_e_\theta)_j, (\Delta\_e_\theta)_j, (Level\_e_\theta)_j\right\}$ – множество параметров входа перехода θ, включающее такие элементы: $(Id\_e_\theta)_j$ – идентификатор входа, $(\delta\_e_\theta)_j$ – локальный счетчик времени, $(\Delta\_e_\theta)_j$ – период простоя, $(Level\_e_\theta)_j$ – уровень представления e_θ.

Ядро входа: $N_\theta^e = \left\{(\rho_\theta)_j, (w_\theta)_j\right\}$, где $(\rho_\theta)_j$ – процедура активации входа; $(w_\theta)_j$ – процедура ожидания. μ_θ – текущая входная метка операторного перехода θ.

Выходы операторного перехода: $(x_\theta)_i = \left\{(o_\theta)_i, (N_\theta^x)_i, (q_\theta^x)_i\right\}$, где $(o_\theta)_i = \left\{(Id\_x_\theta)_i, (\varphi\_x_\theta)_i, (Cur\_x_\theta)_i, (Max\_x_\theta)_i\right\}$ – множество параметров выхода перехода θ, включающее такие элементы: $(Id\_x_\theta)_i$ – идентификатор выхода; $(\varphi\_x_\theta)_i$ – множество допустимых типов меток; $(Cur\_x_\theta)_i$ – текущее количество меток; $(Max\_x_\theta)_i$ – максимально допустимое количество меток.

Ядро выхода: $(N_\theta)_i = \left\{(\gamma_\theta)_i\right\}$, где $(\gamma_\theta)_i$ – процедура деактивации выхода; $(q_\theta^x)_i$ – множество меток на выходе операторного перехода θ.

Ребра сети задают матрицей инцидентности H с элементами:

$$H(p_i, t_j) = \begin{cases} -1, (p_i, t_j) \in \mathbf{F}, \\ +1, (p_i, t_j) \in \mathbf{F}^{-1}, \\ 0, \quad (p_i, t_j) \notin \mathbf{F}, (p_i, t_j) \notin \mathbf{F}^{-1}, \end{cases} \quad 1 \le i \le n, \; 1 \le j \le m.$$

Метки APRO-сети: $\mu_k = \{\lambda_k, \alpha_k\}$, где $\lambda_k = \{Id\_\mu_k, \delta\_\mu_k, \varphi\_\mu_k\}$ – множество параметров метки, включающее такие элементы: Id_μ_k – идентификатор метки, δ_μ_k – время создания метки, φ_μ_k – тип метки, α_k – множество атрибутов метки.

Множество глобальных переменных: $V = (\Delta, \Psi, \Lambda)$, где Δ – подмножество показателей продуктивности, Ψ – подмножество показателей реактивности, Λ – подмножество показателей использования.

3. Принципы функционирования APRO-сетей

Рассмотрим функционирование APRO-сети, адаптировав основные положения семантики последовательных шагов [6] к предложенной версии сетей.

Шагом будем называть мультимножество простых переходов $U = (X, \beta)$, где $X \subseteq T$, $\beta : X \to N$ – функция, задающая отображение X на пространство натуральных чисел N.

Шаг U активируются при условии выполнения для каждой позиции $p_i \in P$ неравенства:

$$M(p_i) \ge \left| \sum_{t_j \in X} H(p_i, \tau_j) \cdot \beta(\tau_j) \right|. \tag{3}$$

Pre-мультимножество шага U объединяет все входные позиции простых переходов, входящих в данный шаг:

$$^\circ U = \left\{ \begin{array}{l} (p_i, \phi_i) \mid p_i \in {}^\bullet\tau_j, \tau_j \in X, \\ \phi_i = \sum_{\tau_j \in X} \beta(\tau_j), 1 \le i \le n, 1 \le j \le m \end{array} \right\}, \tag{4}$$

где ${}^\bullet\tau_j = \left\{p_i \middle| H(p_i, t_j) < 0\right\}$.

Post-мультимножество шага U объединяет все выходные позиции простых переходов, входящих в данный шаг:

$$U^\circ = \left\{ \begin{array}{l} (p_i, \phi_i) \mid p_i \in \tau_j^\bullet, \tau_j \in X, \\ \phi_i = \sum_{\tau_j \in X} \beta(\tau_j), 1 \le i \le n, 1 \le j \le m \end{array} \right\}, \tag{5}$$

где $\tau_j^\bullet = \left\{ p_i \middle| H(p_i, t_j) > 0\right\}$.

При активации и выполнении условий срабатывания переход τ_j может сработать, результатом чего есть изменение количества меток на связанной с данным переходом позиции:

$$M'(p_i) := M(p_i) + H(p_i, \tau_j). \tag{6}$$

В результате срабатывания шага U получим изменение маркирования с $M(U)$ на $M'(U)$ в соответствии с выражением:

$$M'(U) := M(U) + \sum_{p_i \in {}^\circ U} H(p_i, \tau_j) + \sum_{p_i \in U^\circ} H(p_i, \tau_j). \tag{7}$$

Такое изменение маркирования в результате срабатывания шага U часто обозначают $M|U\rangle M'$. Тогда эволюция маркирований σ под воздействием последовательности шагов $\sigma = U_1 \cdots U_k$ может быть представлена в виде:

$$M|\sigma\rangle M' = M|U_1\rangle M_1 \cdots M_{k-1}|U_k\rangle M' \tag{8}$$

При использовании иерархических сетей правила формирования шага должны также учитывать состояния сетей низших уровней. Для этого рассмотрим сложную APRO-сеть, образованную путем суперпозиции APRO-сетей:

$$\hat{\Phi} = \Phi_0(\Phi_1(\cdots\Phi_r(\cdots\Phi g))), \tag{9}$$

где $g \in N$, Φ_r – вложенная сеть, содержащая вложенные сети, то есть использующая операторные переходы.

Сложным шагом $\hat{U}$ на сети $\hat{\Phi}$ будем называть суперпозицию мультимножеств переходов

$$\hat{U} = U_0 \circ U_1 \circ \cdots \circ U_r \circ \cdots \circ U_g ,$$

где $U_r = (T_r, \beta_r)$, $T_r = \left\{ \left(t_j\right)_r \right\}$, $j \le |T_r|$, активирующихся при условии, что для каждой последовательности позиций $\left\langle (p)_0, \cdots (p)_r, \cdots (p)_g \right\rangle$ справедлива последовательность неравенств:

$$M((p)_0) \ge \left| \sum_{(t_j)_0 \in T_0} H\left((p)_0, (t_j)_0\right) \cdot \beta\left((t_j)_0\right) \right|,$$

$$\ldots$$

$$M((p)_r) \ge \left| \sum_{(t_j)_r \in T_r} H\left((p)_r, (t_j)_r\right) \cdot \beta\left((t_j)_r\right) \right|,$$

$$\ldots$$

$$M((p)_g) \ge \left| \sum_{(t_j)_g \in T_g} H\left((p)_g, (t_j)_g\right) \cdot \beta\left((t_j)_g\right) \right|,$$

где $(p)_r$ – произвольная маркированная позиция APRO-сети на уровне r .

Pre-мультимножество сложного шага $\hat{U}$ APRO-сети $\hat{\Phi}$ включает входные позиции переходов разных уровней, входящих в данный шаг:

$$^{\circ}\hat{U} = \left\{ \begin{array}{l} \bigcup\limits_r \left((p_i)_r, (\phi_i)_r\right) \mid p_i \in {}^{\bullet}\left(t_j\right)_r, \\ \left(t_j\right)_r \in T_r,\ (\phi_i)_r = \sum_{(t_j) \in T_r} \beta\left(\left(t_j\right)_r\right), \\ 0 \le r \le q,\ 1 \le i \le |P_r|,\ 1 \le j \le |T_r| \end{array} \right\} . \quad (10)$$

Post-мультимножество сложного шага $\hat{U}$ объединяет все выходные позиции переходов, входящие в данный шаг:

$$\hat{U}^{\circ} = \left\{ \begin{array}{l} \bigcup\limits_r \left((p_i)_r, (\phi_i)_r\right) \mid p_i \in \left(t_j\right)_{\gamma}^{\bullet}, \\ \left(t_j\right)_r \in T_r,\ (\phi_i)_r = \sum_{(t_j) \in T_r} \beta\left(\left(t_j\right)_r\right), \\ 0 \le r \le q,\ 1 \le i \le |P_r|,\ 1 \le j \le |T_r| \end{array} \right\} . \quad (11)$$

Если сработают все переходы, которые входят в шаг $\hat{U}$, получим изменение маркирования с $M(U)$ на $M'(U)$ в соответствии с выражением:

$$M'(\hat{U}) := M(\hat{U}) + \sum_{p \in {}^{\circ}\hat{U}} H(p,t) + \sum_{p \in \hat{U}^{\circ}} H(p,t) , \quad (12)$$

где $t \in \hat{U}$.

Изменение маркирования в результате срабатывания сложного шага $\hat{U}$ будем обозначать $M \left| \hat{U} \right\rangle M'$. Представив последовательность сложных шагов $\hat{\sigma} = \hat{U}_0 \cdots \hat{U}_g$, определим эволюцию сложной APRO-сети $\hat{\Phi}$ в виде выражения:

$$M \left| \hat{\sigma} \right\rangle M' = M \left| \hat{U}_0 \right\rangle M_1 \cdots M_{k-1} \left| \hat{U}_g \right\rangle M' . \quad (13)$$

Таким образом, сложная APRO-сеть является формальным средством описания сложных дискретных систем, характеризующихся параллельным функционированием компонент, связанных не только одноуровневыми связями, а и образовывающих многоуровневые иерархические структуры.

4. Эквивалентность APRO-сетей

Проблема эквивалентности сетей актуальна с момента возникновения сетей Петри в 60-х годах прошлого столетия. За это время сформирован ряд подходов, большинство из которых базируется на применении методики определения достижимости. Такие виды эквивалентностей получили название эквивалентности маркирования и трассовой эквивалентности.

Определение 2. Две APRO-сети Φ_1 и Φ_2 будем считать связанными эквивалентностью маркирования, то есть $Mr(\Phi_1) = Mr(\Phi_2)$, если $P_1 = P_2$, и для каждой позиции p_i, $1 \le i \le n$ существуют одинаковые достижимые маркирования $M_1(p_i) = M_2(p_i)$.

Этот критерий эквивалентности оперирует только достижимыми маркированиями. Поэтому его чаще используют в сетях, характеризующихся отсутствием циклов. Для более сложных сетей важно также определить промежуточную последовательность изменений состояний, ведущих от начального маркирования к достижимому.

Определение 3. Пусть Φ – APRO-сеть, в которой активность каждого перехода $t_j \in T, 1 \le j \le n$ представлена соответствующей функцией активации ρ_j . Множеством трасс $\mathrm{Tr}(\Phi)$ сети Φ будем называть множество последовательностей $\langle \rho_1, \cdots \rho_k \rangle$, если для каждой такой последовательности существует последовательность маркирований $\langle M_1 \cdots M_k \rangle$ такая, что $M_0 \xrightarrow{\rho_1} M_1 \xrightarrow{\rho_2} \cdots \xrightarrow{\rho_k} M_k$.

Две APRO-сети Φ_1 и Φ_2 будем считать связанными трассовой эквивалентностью, если их множества трасс совпадают, то есть $\mathrm{Tr}(\Phi_1) = \mathrm{Tr}(\Phi_2)$.

Как упоминалось ранее, трассовая эквивалентность не универсальна и может использоваться только в случае уверенности в достижимости конечного маркирования M_k для обеих сетей, подлежащих сравнению.

В данном случае возникает возможность применения поведенческой эквивалентности [7], которая базируется не только на требовании эквивалентности трасс, но и предполагает достижение одинаковых маркирований после выполнения соответствующих функций активации.

Определение 4. Отношение R между множествами маркирований двух APRO-сетей является отношением строгого взаимного подобия, если для каждой пары $(M_i, M_j) \in R$ и для произвольной функции активации ρ :

- для срабатывания $M_i \xrightarrow{\rho} M'_i$ существует маркирование M'_j такое, что $M_j \xrightarrow{\rho} M'_j$ при $(M'_i, M'_j) \in R$,

- для срабатывания $M_j \xrightarrow{\rho} M'_j$ существует маркирование M'_i такое, что $M_i \xrightarrow{\rho} M'_i$ при $(M'_i, M'_j) \in R$.

Две APRO-сети Φ_1 и Φ_2 строго взаимно подобны, если существует отношение строгого взаимного подобия R , содержащее пару (M_{01}, M_{02}) начальных маркирований сетей Φ_1 и Φ_2 .

Следует отметить, что доказательство данных критериев эквивалентности, к сожалению, ведет к возникновению NP-полных алгоритмов. Причина такой сложности заключается в том, что доказательство трассовой эквивалентности нуждается в решении проблемы достижимости. Но на сегодня данная проблема не имеет обобщенного решения даже для базовых сетей Петри.

Поскольку APRO-сети не ограничены жесткими правилами срабатывания переходов, то существование трассы определяется по наличию связи между соответствующими позициями и алгоритмическими свойствами переходов. Этот факт значительно затрудняет формализацию проблемы достижимости и делает ее зависимой от информационного наполнения атрибутов меток. Поэтому алгоритмический подход пока рассматривается как единственный способ для установления строгого взаимного подобия APRO-сетей.

Суть алгоритма установления строгого взаимного подобия APRO-сетей заключается в поиске хотя бы одной пары маркирований $(M_i)_1 \nsim (M_i)_2$, которая бы опровергала строгое взаимное подобие сетей Φ_1 и Φ_2. Если этот факт не доказан, то следующий шаг состоит в доказательстве $(M_i)_1 \sim (M_i)_2$.

Пусть $((M_i)_1,(M_i)_2)$ – произвольная пара маркирований сетей Φ_1 и Φ_2. Тогда алгоритм определения строгого взаимного подобия этих маркирований состоит из следующих шагов:

1. Выполнить попытку перехода $(M_i)_1 \xrightarrow{\rho} (M'_i)_1$ от маркирования $(M_i)_1$ к произвольному маркированию $(M'_i)_1$.
2. При условии успешного выполнения перехода $(M_i)_1 \xrightarrow{\rho} (M'_i)_1$ выполнить переход $(M_j)_2 \xrightarrow{\rho} (M'_j)_2$.
3. Если успешно выполнены переходы $(M_i)_1 \xrightarrow{\rho} (M'_i)_1$ и $(M_j)_2 \xrightarrow{\rho} (M'_j)_2$, установить новую начальную пару маркирований путем переприсваиваний: $(M_i)_1 := (M'_i)_1$ и $(M_j)_2 := (M'_j)_2$.

Алгоритм установления строгого взаимного подобия всегда начинается с начальной пары маркирований $((M)_1,(M)_2)$ и может развиваться в нескольких направлениях в зависимости от результата первого шага. Второй шаг всегда зависит от первого, поскольку на первом и втором шаге переходы возникают под воздействием одной и той же процедуры активации. Алгоритм функционирует до момента достижения конечной пары маркирований. Остановка его происходит при условии невозможности срабатывания перехода на первом или на втором шаге обработки текущей пары маркирований. Если алгоритм останавливается на первом этапе обработки текущей пары маркирований, то считают, что $(M_i)_1 \sim (M_i)_2$, а в случае остановки на втором шаге $-(M_i)_1 \nsim (M_i)_2$. В случае, если всегда возможно выполнить переход как на первом, так и на втором шаге, считают, что $(M_i)_1 \sim (M_1)_2$.

5. Выводы

В результате стремительного развития современных технологий окружающий нас мир наполнился большим количеством объектов, которые могут быть классифицированы как сложные дискретные системы. Развитие и совершенствование таких систем требует поиска новых подходов к созданию формальных средств их описания.

Новизна предложенной версии APRO-сетей состоит в развитии свойств известных формальных сетевых описаний путем введения дополнительных структурных элементов, строгом определении структур данных, ассоциируемых с этими структурными элементами, и создания правил функционирования сети на базе семантики последовательных шагов. К новым структурным элементам APRO-сети следует отнести операторные переходы, входы операторных переходов и выходы операторных переходов. Введение этих элементов позволило формализовать механизм описания иерархических моделей с произвольной степенью вложенности.

Позиции APRO-сети содержат структуры данных, разделенные на две основные категории. Первая категория объединяет параметры позиций, а вторая включает упорядоченные структуры хранения меток. Такой подход позволил создать формальное описание, инвариантное к информационному наполнению рабочей нагрузки. Подобное разделение структур данных переходов на параметры и ядро способствовало повышению независимости формального описания от алгоритмического наполнения модели. При использовании конкретных алгоритмов процедур переходов и задании конкретных структур данных для атрибутов меток APRO-сети можно адаптировать к описанию широкого круга сложных дискретных систем с возможностью обработки имитационной моделью, полученной по данному описанию, реальной рабочей нагрузки.

Адекватность представления объектов моделирования с помощью APRO-сетей базируется на определениях трассовой и поведенческой эквивалентности. Предложен алгоритм определения строгого взаимного подобия для APRO-сетей.

Формальное описание APRO-сетей может быть положено в основу языков описания сложных дискретных систем при разработке программных средств имитационного моделирования.

ЛИТЕРАТУРА

1. Трахтенброт Б.А., Барздинь Я.М. Конечные автоматы. Поведение и синтез.− М.: Наука, 1970.− 400 с.
2. Бусленко Н.П. Моделирование сложных систем.− М.: Наука, 1978.− 400 с.
3. Reising W. Petri nets: an introduction. − New York: Springer-Verlag, 1985. −161 p.
4. Noe J., Nutt G. I. Macro E-nets for representation of parallel systems // IEEE Trans. on Computers.– 1973.– vol.C-22, № 8.– P.718–727.
5. Нестеренко Б.Б., Новотарский М.А. Мультипроцессорные системы.− К.: Институт математики АН Украины, 1995.− 408 с.
6. Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE.– 1981.– vol.77, № 4.− P.541-580.
7. Jancar P. Decidability Questions for Bisimilarity of Petri Nets and Some Related Problems: Technical Report ECS-LFCS-93-261 / Dept. of Computer Science, University of Ostrava.– Ostrava, 1993.– 24 p.

пост. 29.05.07.

