ПРОГРАММИРОВАНИЕ, 2001, № 6, с. 30–43

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

УДК 681.322

КОМПОЗИЦИОНАЛЬНЫЙ ПОДХОД К РАЗРАБОТКЕ ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЕННЫХ СИСТЕМ НА ОСНОВЕ СЕТЕЙ ПЕТРИ*

© 2001 г. Н. А. Анисимов, Е. А. Голенков, Д. И. Харитонов
Институт автоматики и процессов управления Дальневосточного отделения РАН
690041 Владивосток, ул. Радио, 5
E-mail: demiurg@iacp.vl.ru
Поступила в редакцию 12.01.2000 г.

В работе предлагается формализм, основанный на теории сетей Петри и предназначенный для применения в композициональном подходе к разработке и анализу сложных параллельных и распределенных систем. Вводится понятие мультипометки сетей Петри, где переход может быть помечен не только одним символом, но и мультимножеством символов. Вводятся операции над мультипомеченными сетями Петри – параллельная композиция и ограничение. На базе понятия мультипометки дается определение объекта сети Петри. Объект сети Петри определяется как сеть Петри с множеством мультипометок, где каждая мультипометка трактуется как точка доступа к объекту. Вводится операция композиции объектов. На основе бисимуляционной эквивалентности сетей Петри вводится понятие эквивалентности объектов. Показано, что эквивалентность конгруэнтна по отношению к операции композиции объектов. Также показывается, что операция композиции обладает свойствами коммутативности и ассоциативности.

1. ВВЕДЕНИЕ

Известно, что формальный подход играет важную роль в разработке параллельных и распределенных систем. В то же время для практической применимости формальных аппаратов необходимо, чтобы они обладали тремя ключевыми свойствами – *параллельностью*, *структурностью* и *абстрактностью* [1]. К сожалению, ни один из существующих формальных аппаратов не обладает в полной мере этими свойствами.

Действительно, абстрактные параллельные алгебраические языки, такие как CCS [2], TCSP [3], ACP [4] и их практические версии – языки LOTOS [5], `occam` [6] и PSF [7], хотя и обладают свойствами абстракции и структуризации, но, однако, не позволяют представлять параллелизм достаточно адекватным способом. В частности, они трактуют параллельность двух событий как всевозможные их последовательные исполнения, так называемый интерливинг. Так, например, параллельность двух событий a и b определяется как выполнение либо последовательности ab либо последовательности ba, что, безусловно, является ограниченным пониманием параллельности хотя бы потому, что исключает одновременное выполнение событий.

Другая группа формализмов, которая принадлежит к теории сетей Петри [8, 9], позволяет представить параллельность более адекватным и естественным способом. Однако эти формализмы страдают отсутствием свойств абстрактности и структурности, что препятствует их широкому применению при решении практических задач. В последние годы для преодоления этих недостатков проведены серьезные исследования. Часть из них посвящена замене интерливинговой семантики алгебраических языков на семантику так называемой "истинной" параллельности, основанной на сетевых моде-

*Работа выполнена при поддержке Миннауки РФ, проект 0201.01.225 "Разработка средств повышения эффективности программ для вычислительных систем параллельной архитектуры".

лях. Стоит упомянуть такие попытки для CCS [10, 11], TCSP [1, 11], ACP [12], а также для языков LOTOS [13] и `occam` [6]. Следует отметить, что эти подходы встретили серьезные трудности, так как абстрактные параллельные алгебраические языки изначально разрабатывались вне сетевой семантики. В частности, все они обладают выразительной мощностью машины Тьюринга, в то время как сети Петри имеют меньшую мощность [8]. Это несоответствие вынуждает искать подмножества абстрактных языков, семантика которых могла бы быть выражена в терминах сетей Петри [10, 13].

Другой подход к этой проблеме включает работы по внесению структурности и абстрактности непосредственно в сети Петри, безотносительно к каким-либо языкам. Работы в этом направлении содержат попытки усиления композициональных возможностей сетей Петри, заключающиеся в определения понятия интерфейса для сети Петри и набора правил их композиции [14]. Часть работ в качестве такого интерфейса использовала места [15–17], а другая часть – переходы [18–20]. Композиция сетей заключалась соответственно в слиянии либо мест, либо переходов. Наиболее значительный прогресс в этом направлении был достигнут в работах, выполняемых в рамках проекта Caliban и его предшественника проекта DEMON [21] Европейской программы Esprit. Результатом этих исследований является исчисление PBC (Petri Box Calculus), изложенное в [22].

По существу, модель PBC, названная термином *Petri box*, есть сеть Петри с интерфейсом, определенным как функция пометки. Эта функция сопоставляет переходу сети мультимножество элементарных действий. Оператор синхронизации сетей, используя эту информацию, генерирует множество переходов синхронизации, сливая взаимодействующие переходы. Предполагается, что элементарное действие – это имя канала взаимодействия, который может иметь набор параметров.

Следующим шагом в этом направлении является дальнейшая структуризация событий, позволяющая выделить направления взаимодействия более явно. Этот подход учитывает природу распределенных систем, где разработчик должен специфицировать модули системы и их взаимодействия. Именно в этом подходе было введено понятие объекта сети Петри, которое является обобщением помеченных сетей Петри. Взамен единственной пометочной функции берется несколько функций, получивших название точек доступа. Каждая точка доступа сопоставляет каждому переходу мультимножество имен, где каждое такое имя может трактоваться как элементарная операция взаимодействия с другими объектами. По сравнению с PBC, направления взаимодействия выделяются явно, что ведет к "расщеплению" одной пометочной функции на несколько.

Впервые понятие объекта сети Петри (без мультипометки) было введено в работе [23] и впоследствии развито в [24, 25], где, однако, основное внимание было уделено применению аппарата к области разработки протоколов вычислительных сетей. Настоящая же работа посвящена дальнейшему развитию аппарата объектов сетей Петри, его строгому формальному определению и, в частности, обобщению на случай мультипометки.

Следует привести некоторые пояснения относительно выбора термина для предложенной модели. Дело в том, что изначально данный аппарат разрабатывался как формальная модель протокольного объекта, являющегося одним из ключевых понятий архитектуры вычислительных сетей[1] [26], для которого используется английский термин *entity*. В отечественной литературе для обозначения этого понятия укоренилось использование термина *объект*, что пока не вызывает особых недоразумений и путаницы с понятием объекта из области объектно-ориентированных технологий программирования. Авторы выражают надежду, что используемый термин *объект сети Петри* также не вызовет путаницы с объектно-ориентированными технологиями и, в частности, с объектно-ориентированными моделями теории сетей Петри [27].

2. МУЛЬТИПОМЕТКА СЕТЕЙ ПЕТРИ

Пусть $A = \{a_1, a_2, \ldots, a_k\}$ есть некоторое множество. Мультимножество на множестве A есть функция $\mu : A \to \{0, 1, 2, \ldots\}$, которая ассоциирует с каждым элементом множества A неотрицательное целое число. Иногда удобно записывать мультимножество на множестве A как формальную сумму $n_1 a_1 + n_2 a_2 + \ldots + n_k a_k$

[1] Или, более точно, Эталонной модели взаимодействия открытых систем Международной организации по стандартизации.

или $\sum n_i a_i$, где $n_i = \mu(a_i)$ есть число вхождений $a \in A$ в мультимножество. Как правило, при записи суммы ее элементы с $a_i = 0$ будут опускаться. Объединение и пересечение двух мультимножеств $\mu_1 = n_1a_1 + \ldots + n_ka_k$ и $\mu_2 = m_1a_1 + \ldots + m_ka_k$, определенных на множестве A, определяется соответственно как $\mu_1 + \mu_2 = (n_1 + m_1)a_1 + \ldots + (n_k + m_k)a_k$ и $\mu_1 - \mu_2 = (n_1 - m_1)a_1 + \ldots + (n_k - m_k)a_k$, где последняя операция определяется только тогда, когда $n_i - m_i \geq 0$ для всех $1 \leq i \leq k$. Мы будем писать $\mu_1 \leq \mu_2$, если $n_i \leq m_i$ для каждого $1 \leq i \leq k$, и $\mu_1 < \mu_2$, если $\mu_1 \leq \mu_2$ и $\mu_1 \neq \mu_2$. Если $n_i = 0$ для всех i, тогда такое мультимножество будет обозначаться как $\mathbf{0}$. Мы также будем писать $a \in \mu$, если $\exists n > 0 : (a, n) \in \mu$.

Множество всех конечных мультимножеств на множестве A будет обозначаться как $\mathcal{M}(A)$. Множество всех возможных последовательностей, составленных из символов множества A, включая пустую строку ϵ, будет обозначаться как A^*.

Пусть $f : A \to B$ есть некоторая функция, и $X \subseteq A$. Тогда $f\lceil X$ есть проекция функции f на X, определяемая как $f\lceil X = \{(a, b) \in f \mid a \in X\}$.

Определение 2.1. *Сеть Петри.*

Сеть Петри определяется как набор $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$, где

1. S есть конечное множество *мест*;
2. T есть конечное множество *переходов* такое, что $S \cap T = \emptyset$;
3. ${}^\bullet() : T \to \mathcal{M}(S)$ есть входная функция инцидентности;
4. $()^\bullet : T \to \mathcal{M}(S)$ есть выходная функция инцидентности;
5. $M_0 \in \mathcal{M}(S)$ есть начальная маркировка.

■ 2.1

Мультимножества ${}^\bullet t$ и $t^\bullet$ называются входным и выходным множествами перехода $t \in T$, соответственно. Функции ${}^\bullet()$ и $()^\bullet$ могут быть естественно расширены на мультимножества: ${}^\bullet(n_1t_1 + \ldots + n_kt_k) = n_1{}^\bullet t_1 + \ldots + n_k{}^\bullet t_k$ и $(n_1t_1 + \ldots + n_kt_k)^\bullet = n_1t_1^\bullet + \ldots + n_kt_k^\bullet$.

Далее будет использоваться привычное графическое представление сети Петри как двудольного графа, где места изображаются кружками, а переходы – прямоугольниками. Места и переходы соединяются ориентированными дугами, представляя входные и выходные функции инцидентности. Вес дуг задается целыми числами, расположенными около дуг. Маркировка изображается метками, помещенными внутрь мест.

В данной работе будет использоваться шаговая семантика сетей Петри, основанная на срабатывании последовательностей мультимножеств переходов.

Маркировка M сети $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$ есть мультимножество, заданное на S, т.е. $M \in \mathcal{M}(S)$. Будем говорить, что шаг (мультимножество переходов) $\Theta \in \mathcal{M}(T)$ возбужден в маркировке M, если ${}^\bullet\Theta \leq M$. Шаг $\Theta \in \mathcal{M}(T)$, возбужденный в маркировке M, может сработать, приводя к новой маркировке M', что записывается как $M[\Theta\rangle M'$, где $M' = M - {}^\bullet\Theta + \Theta^\bullet$. Следует отметить, что, если шаг Θ возбужден в маркировке M, то шаг $\Theta' < \Theta$ также возбужден при этой маркировке M. Если $\Phi = \Theta_1\Theta_2\ldots\Theta_n \in (\mathcal{M}(T))^*$, то $M[\Phi\rangle M'$ означает, что существует последовательность $M_1, M_2, \ldots, M_{n-1}$ такая, что $M[\Theta_1\rangle M_1[\Theta_2\rangle \ldots M_{n-1}[\Theta_n\rangle M'$. В этом случае будем говорить, что M' достижима из M. $M[\rangle M'$ означает, что существует $\Phi \in \mathcal{M}(T)^*$ такое, что $M[\Phi\rangle M'$. Множество всех достижимых маркировок из M определяется как $[M\rangle = \{M' \mid M[\rangle M'\}$. Если мы желаем указать сеть, для которой формулируется некоторое утверждение, мы будем указывать соответствующий префикс. Например, $\Sigma : M[\Phi\rangle M'$, $\Sigma_1 : M \leq {}^\bullet\Theta$.

Пусть Δ есть конечный алфавит имен, и $\overline{\Delta} = \{\overline{a} \mid a \in \Delta\}$ есть связанный с ним алфавит дополнительных имен. Другими словами, мы определяем биекцию $\bar{\cdot} : \Delta \to \overline{\Delta}$, которая означает взаимнооднозначное соответствие между именами и их дополнительными именами. Для простоты функция, обратная функции $\bar{\cdot}$, будет также записываться как $\bar{\cdot} : \overline{\Delta} \to \Delta$. Таким образом, мы имеем $\overline{\overline{a}} = a$. Пусть $\tau \notin \mathcal{M}(\Delta \cup \overline{\Delta})$ есть специальный символ, который ассоциируется с невидимым действием. Далее будем обозначать $Vis = \Delta \cup \overline{\Delta}$ и $Act = \mathcal{M}(Vis) \cup \{\tau\}$. Функция $\bar{\cdot}$ может быть расширена на мультимножество имен $\overline{n_1a_1 + \ldots + n_ka_k} = n_1\bar{a}_1 + \ldots + n_k\bar{a}_k$. Например,

$$\overline{\bar{a} + 2a + 3\bar{b} + c} = a + 2\bar{a} + 3b + \bar{c}.$$

Определение 2.2. *Пометка.*

Пометка сети Петри $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0\rangle$ есть набор $\lambda = \langle \Delta, \sigma\rangle$, где Δ есть некоторый алфавит, и $\sigma : T \to Act$ есть пометочная функция. ■ 2.2

Это определение является расширением хорошо известного определения пометочной функции. В частности, каждый переход может быть помечен не только одним символом, но и их мультимножеством. Переход, помеченный символом τ, будет трактоваться как невидимый или внутренний переход.

Физическим смыслом метки является элементарное взаимодействие. Так, например, будем считать, что символ $a \in \Delta$ соответствует посылке сообщения с именем a, а дополнительный символ $\overline{a} \in \overline{\Delta}$ – приему сообщения с именем a. Символ τ не связан с взаимодействием и характеризует некоторое внутреннее событие. Мультимножество символов, помечающее переход, соответствует одновременному приему и/или передаче сообщений, происходящих при срабатывании этого перехода. Так, например, если $\sigma(t) = \overline{a} + 2b$, то срабатыванию перехода t соответствует одновременный прием сообщения a и посылка двух сообщений b.

Пометочная функция σ расширяется на мультимножество переходов, $\sigma : \mathcal{M}(T) \to Act$, следующим образом. Если $\Theta = \sum n_i a_i \in \mathcal{M}(T)$, то $\sigma(\Theta) = \sum n_i \sigma(a_i)$. Если каждый переход t в шаге Θ невидим, т.е. $\sigma(t) = \tau$, тогда мы будем писать $\sigma(\Theta) = \tau$.

Функция σ может быть также естественно расширена до гомоморфизма $\sigma : \Big(\mathcal{M}(T)\Big)^* \to \to \Big(\mathcal{M}(Act)\Big)^*$. Определим функцию $\sigma^+ : (\mathcal{M}(T))^* \to (Vis)^*$, которая удаляет все τ-символы из последовательностей:

$$\sigma^+(\Theta) = \begin{cases} \epsilon, & \text{если } \sigma(\Theta) = \tau; \\ \sigma(\Theta), & \text{в противном случае;} \end{cases}$$

$$\sigma^+(\Phi\Theta) = \sigma^+(\Phi)\sigma^+(\Theta).$$

Далее буквой Θ будет обозначаться шаг, а буквой Φ – последовательность шагов.

Если $W \in (\mathcal{M}(Vis))^*$ и $\lambda = \langle \Delta_\lambda, \sigma_\lambda \rangle$ есть пометки сети Петри Σ, тогда $M(W\rangle_\lambda M'$ означает, что $\Phi \in (\mathcal{M}(T))^* : M[\Phi\rangle M'$ и $\sigma^+_\lambda(\Phi) = W$. Далее для краткости последовательность $M\ [\Phi\rangle\ M'$ с $\sigma(\Phi) = \epsilon$ будет записываться как $M \Longrightarrow M'$, а сам символ $\Longrightarrow$ будет обозначать невидимую последовательность шагов.

Определение 2.3. *Ограничение.*

Пусть дана сеть $\Sigma_1 = \langle S_1, T_1, ^\bullet()_1, ()^\bullet_1, M_{01} \rangle$ и ее пометка $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$. Тогда ограничение сети Σ по отношению к α есть новая сеть $\Sigma = \partial_\alpha(\Sigma_1)$ такая, что

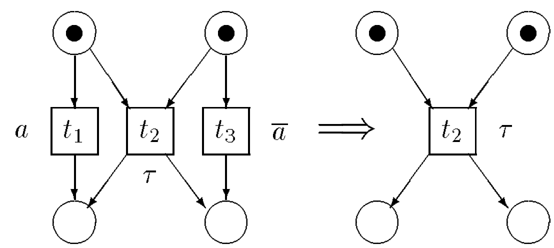

Рис. 1. Пример применения операции ограничения.

1. $S = S_1$;
2. $T = T_1 \setminus \{t \in T \mid \sigma_\alpha(t) \neq \tau\}$;
3. $^\bullet(t) =^\bullet (t)_1$, $t \in T$;
4. $(t)^\bullet = (t)^\bullet_1$, $t \in T$;
5. $M_0 = M_{01}$.

■ 2.3

Менее формально, ограничение сети удаляет каждый переход, помеченный именем из Vis вместе с прилегающими дугами. На рис. 1 иллюстрируется применение этой операции. Здесь переходы t_1 и t_3, помеченные именами из Vis, удаляются, а невидимый τ-переход остается.

Предположим, что для сети $\Sigma = \partial_\alpha(\Sigma_1)$ определены две пометки $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$ и $\beta = = \langle \Delta_\beta, \sigma_\beta \rangle$. Пометка β для сети Σ может быть естественно ограничена следующим образом: $\tilde{\beta} = \langle \Delta_\beta, \sigma_\beta \lceil T \rangle$. Иногда, когда это не приводит к недоразумениям, будет записываться β вместо $\tilde{\beta}$.

Предложение 2.4. *Пусть Σ есть сеть, и α и β – ее пометки. Справедливо равенство* $\partial_{\tilde{\alpha}}(\partial_\beta(N)) = \partial_{\tilde{\beta}}(\partial_\alpha(N))$.

Доказательство. Разделим T на четыре непересекающихся подмножества: $T = T_{11} \cup T_{12} \cup \cup T_{21} \cup T_{22}$, где

$$\begin{aligned} T_{11} &= \{t \in T \mid \sigma_\alpha(t) \neq \tau \neq \sigma_\beta(t)\}, \\ T_{12} &= \{t \in T \mid \sigma_\alpha(t) \neq \tau = \sigma_\beta(t)\}, \\ T_{21} &= \{t \in T \mid \sigma_\alpha(t) = \tau \neq \sigma_\beta(t)\}, \\ T_{22} &= \{t \in T \mid \sigma_\alpha(t) = \tau = \sigma_\beta(t)\}, \end{aligned}$$

$$\begin{aligned} \partial_{\tilde{\alpha}}(\partial_\beta(\Sigma)) &= \partial_{\tilde{\alpha}}(\langle S, T \setminus T_{12} \cup T_{22}), \\ & \quad ^\bullet()\lceil(T_{11} \cup T_{21}), \\ & \quad ()^\bullet\lceil(T_{11} \cup T_{21}), M_0\rangle) = \\ &= \langle S, T_{11}, ^\bullet()\lceil T_{11}, ()^\bullet\lceil T_{11}, M_0\rangle; \end{aligned}$$

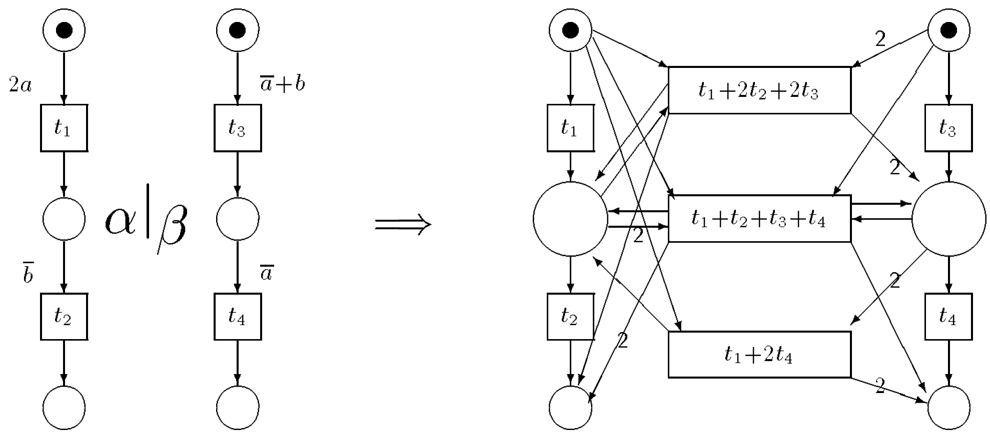

Рис. 2. Параллельная композиция сетей Петри.

$$\begin{aligned}\partial_{\tilde{\beta}}(\partial_\alpha(\Sigma)) &= \partial_{\tilde{\beta}}(\langle S, T\setminus T_{21}\cup T_{22}), \\ &\ {}^\bullet()\lceil(T_{11}\cup T_{12}), \\ &\ ()^\bullet\lceil(T_{11}\cup T_{12}), M_0\rangle) = \\ &= \langle S, T_{11}, {}^\bullet()\lceil T_{11}, ()^\bullet\lceil T_{11}, M_0\rangle.\end{aligned}$$

■ 2.4

Это предложение позволяет обобщить операцию ограничения на множество пометок.

Определение 2.5. *Обобщенное ограничение.*

Пусть Σ есть сеть Петри и $H = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ есть множество ее пометок. Тогда $\partial_H(\Sigma) = \partial_{\alpha_1}\circ\partial_{\alpha_2}\circ\ldots\circ\partial_{\alpha_n}(\Sigma)$. ■ 2.5

3. ПАРАЛЛЕЛЬНАЯ КОМПОЗИЦИЯ МУЛЬТИПОМЕЧЕННЫХ СЕТЕЙ ПЕТРИ

Операция параллельной композиции играет важную роль в композициональном подходе, так как позволяет конструировать системы с взаимодействующими компонентами. Существует ряд определений этой операции для сетей Петри, см. [10–12, 16]. Все эти операции определены для взаимно однозначных взаимодействий, т.е. когда переход одной сети сливается не более чем с одним переходом другой сети. Нам, однако, необходима обобщенная параллельная композиция на случай мультивзаимодействия в соответствии с определением мультипометки.

Пусть $\Sigma_1 = \langle S_1, T_1, {}^\bullet()_1, ()_1^\bullet, M_{01}\rangle$ и $\Sigma_2 = \langle S_2, T_2, {}^\bullet()_2, ()_2^\bullet, M_{02}\rangle$ – две сети Петри. Пусть также сети Σ_1 и Σ_2 имеют непересекающиеся множества мест и переходов, т.е. $S_1\cap S_2 = T_1\cap T_2 = \emptyset$.

Определение 3.1. *Параллельная композиция мультипомеченных сетей Петри.*

Пусть Σ_1 и Σ_2 – сети Петри с пометками $\alpha = \langle\Delta_\alpha, \sigma_\alpha\rangle$ и $\beta = \langle\Delta_\beta, \sigma_\beta\rangle$, соответственно. Параллельная композиция сетей Σ_1 и Σ_2 по отношению к α и β есть новая сеть $\Sigma = (\Sigma_1\ {}_\alpha|_\beta\ \Sigma_2)$ такая, что

1. $S = S_1\cup S_2$,
2. $T = T_1\ \cup\ T_2\ \cup\ T_1\ {}_\alpha\otimes_\beta\ T_2$, где
$$\begin{aligned}T_1\ {}_\alpha\otimes_\beta\ T_2 \ = \{\ & \mu_1+\mu_2 \mid \mu_1\in\mathcal{M}(T_1), \\ & \mu_2\in\mathcal{M}(T_2), \\ & \tau\notin\sigma_\alpha(\mu_1) = \overline{\sigma_\beta(\mu_2)}, \\ & \text{сумма}\ \ \mu_1+\mu_2 \\ & \text{минимальна}\ \},\end{aligned}$$
3. ${}^\bullet() = {}^\bullet()_1 \cup {}^\bullet()_2 \cup \{(\mu_1+\mu_2, {}^\bullet(\mu_1)_1 + {}^\bullet(\mu_2)_2) \mid \mu_1+\mu_2\in T, \mu_1\in\mathcal{M}(T_1), \mu_2\in\mathcal{M}(T_2)\}$,
4. $()^\bullet = ()_1^\bullet\cup()_2^\bullet\cup\{(\mu_1+\mu_2, (\mu_1)_1^\bullet + (\mu_2)_2^\bullet) \mid \mu_1+\mu_2\in T, \mu_1\in\mathcal{M}(T_1), \mu_2\in\mathcal{M}(T_2)\}$,
5. $M_0 = M_{01} + M_{02}$.

Сумма $\mu_1+\mu_2$ минимальна, если не существует суммы $\mu_1'+\mu_2'$ такой, что $\mu_1'+\mu_2' < \mu_1+\mu_2$ и $\sigma_\alpha(\mu_1) = \overline{\sigma_\beta(\mu_2)}$. ■ 3.1

Менее формально, две сети Σ_1 и Σ_2 просто объединяются с добавлением новых переходов синхронизации $T_1\ {}_\alpha\otimes_\beta\ T_2$. Эти новые переходы задаются мультимножествами символов $\mu_1+\mu_2$, $\mu_1\in\mathcal{M}(T_1)$, $\mu_2\in\mathcal{M}(T_2)$. Для новых переходов вычисляются их входные и выходные мультимножества: ${}^\bullet(\mu_1+\mu_2) = {}^\bullet(\mu_1) + {}^\bullet(\mu_2)$,

$(\mu_1 + \mu_2)^\bullet = (\mu_1)^\bullet + (\mu_2)^\bullet$. Иногда, когда ясно из контекста, будет записываться $T_1 \otimes T_2$ взамен $T_1 \ {}_\alpha\!\otimes_\beta \ T_2$.

На рис. 2 показан пример параллельной композиции. В этом примере добавляются три перехода синхронизации. Первый переход – $(t_1 + 2t_2) + 2t_3$, где $\sigma_\alpha(t_1 + 2t_2) = \overline{\sigma_\beta(2t_3)} = 2a + 2\bar{b}$, второй – $(t_1 + t_2) + (t_3 + t_4)$ с $\sigma_\alpha(t_1 + t_2) = \overline{\sigma_\beta(t_3 + t_4)} = 2a + \bar{b}$, и третий – $t_1 + 2t_4$ с $\sigma_\alpha(t_1) = \overline{\sigma_\beta(2t_4)} = 2a$. Если, например, переименовать t_4 в, скажем, τ, то тогда второй и третий переходы синхронизации не будут сформированы.

Очевидно, что введенная операция параллельной композиции будет практически полезна, если мы снабдим ее процедурой для вычисления множества переходов синхронизации $T_1 \otimes T_2$. Оказывается, что эта проблема может быть сведена к хорошо известной проблеме нахождения множества минимальных инвариантов для сети Петри [28].

Теорема 3.2. *Нахождение переходов синхронизации.*

Проблема нахождения множества переходов синхронизации для параллельной композиции сетей Петри сводима к проблеме нахождения наименьшего множества инвариантов сети Петри.

Доказательство. Пусть $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$ и $\beta = \langle \Delta_\beta, \sigma_\beta \rangle$ есть пометочные функции сетей Петри Σ_1 и Σ_2, соответственно. Пусть $\Delta = \Delta_\alpha \cup \Delta_\beta$, $T_1 = \{t_1^1, t_2^1, \ldots, t_n^1\}$, $T_2 = \{t_1^2, t_2^2, \ldots, t_m^2\}$. Построим новую сеть Петри следующим способом: $\tilde{\Sigma}_{12} = \langle \Delta, T_1 \cup T_2, {}^\bullet(), ()^\bullet, M_0 \rangle$, где $M_0 = \mathbf{0}$,

$$(t)^\bullet = \begin{cases} \sigma_\alpha^+(t)\lceil\Delta, & \text{если } t \in T_1; \\ \sigma_\beta^+(t)\lceil\Delta, & \text{если } t \in T_2; \end{cases}$$

$${}^\bullet(t) = \begin{cases} \overline{\sigma_\alpha^+(t)}\lceil\Delta, & \text{если } t \in T_1; \\ \overline{\sigma_\beta^+(t)}\lceil\Delta, & \text{если } t \in T_2. \end{cases}$$

Здесь проекция $\sigma^+(t)\lceil\Delta$ оставляет только прямые имена из Δ. Заметим, что срабатывание перехода t_i^1 добавляет $\sigma_\alpha^+(t_i^1)\lceil\Delta$ и вычитает $\overline{\sigma_\alpha^+(t_i^1)}\lceil\Delta$ из текущей маркировки. Срабатывание же перехода t_j^2 добавляет $\sigma_\beta^+(t)\lceil\Delta$ и вычитает $\overline{\sigma_\beta^+(t)}\lceil\Delta$ из текущей маркировки. Поэтому для того, чтобы вновь достичь нулевой маркировки срабатыванием последовательности $v \in (T_1 \cup T_2)^*$, $M_0[v\rangle M_0$, должно выполняться условие $\sigma_\alpha(x) = \overline{\sigma_\beta(y)}$. Здесь x и y – мультимножества переходов из T_1 и T_2, встречающиеся в v. С другой стороны, известно, что, если существует $v \in (T_1 \cup T_2)^* : M_0[v\rangle M_0$, то характеристический вектор v есть Т-инвариант. Более того, если $x + y$ минимально, то характеристический вектор мультимножества $x + y$ есть минимальный Т-инвариант сети $\tilde{\Sigma}_{12}$. Далее из полученного множества удаляются инварианты, в которые не входят переходы либо из T_1, либо из T_2. Последняя операция необходима из-за того, что нас интересуют только взаимодействия между сетями, но не автовзаимодействия, где сеть может взаимодействовать сама с собой. ■ 3.2

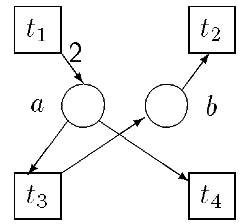

Рис. 3. Пример построения сети $\tilde{\Sigma}_{12}$.

Таким образом, для генерации множества переходов синхронизации $T_1 \otimes T_2$ необходимо построить сеть Σ_{12} и вычислить ее множество минимальных Т-инвариантов. Каждый такой инвариант $f = \langle f_1, \ldots, f_{n+m} \rangle$ даст один переход синхронизации $f_1 t_1^1 + \ldots + f_n t_n^1 + f_{n+1} t_1^2 + \ldots + f_{n+m} t_m^2$. Существует множество алгоритмов нахождения Т-инвариантов для сетей Петри, основанные на алгоритме Фаркаса (Farcas). Некоторые из них представлены в работе [28]. В общем случае, вычисление множества $T_1 \otimes T_2$ является сложной задачей, имеющей экспоненциальную сложность, хотя для ее упрощения могут быть использованы некоторые эвристики (см. [29]).

Возвращаясь к примеру, изображенному на рис. 2, можно построить сеть $\tilde{\Sigma}_{12}$ (см. рис. 3). Нетрудно убедиться, что сеть $\tilde{\Sigma}_{12}$ имеет три минимальных инварианта: $t_1 + 2t_4$, $t_1 + 2t_2 + 2t_3$ и $t_1 + t_2 + t_3 + t_4$.

Далее необходимо определить расширение пометки γ сети Σ_1 на сеть $(\Sigma_1 \ {}_\alpha|_\beta \ \Sigma_2)$, которая будет обозначаться как $\tilde{\gamma}$.

Определение 3.3. *Расширение пометочной функции.* Пусть $\alpha = \langle \Delta_\alpha, \sigma_\alpha \rangle$ и $\gamma = \langle \Delta_\gamma, \sigma_\gamma \rangle$ – две пометки сети Σ_1, и $\beta = \langle \Delta_\beta, \sigma_\beta \rangle$ – пометка сети Σ_2. Тогда расширение γ на сеть $(\Sigma_1 \ {}_\alpha|_\beta \ \Sigma_2)$

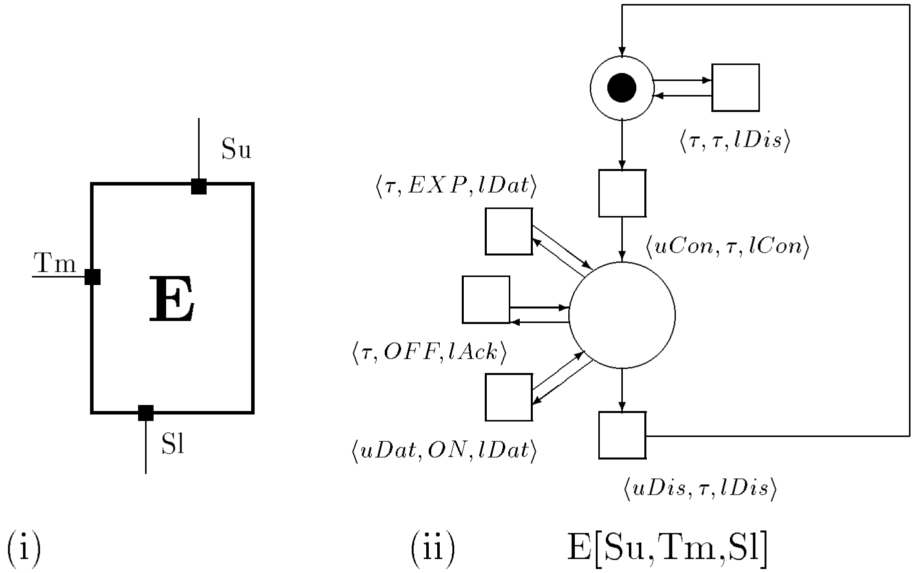

Рис. 4. Пример объекта.

есть пометка $\tilde{\gamma} = \langle \Delta_\gamma, \tilde{\sigma}_\gamma \rangle$, где

$$\begin{aligned} \tilde{\sigma}_\gamma = \sigma_\gamma \quad & \cup \quad \{(t, \tau) \mid t \in (T_2)\} \cup \\ & \cup \quad \{(x + y, \sigma_\gamma(x)) \mid x + y \in T_1 \otimes T_2, \\ & \qquad x \in \mathcal{M}(T_1), y \in \mathcal{M}(T_2)\}. \end{aligned}$$

■ 3.3

Другими словами, пометка переходов из T_1 не меняется, пометка перехода синхронизации $t \in T_1 \otimes T_2$ в γ определяется как сумма пометок составляющих переходов, и переходы из T_2 помечаются символом τ.

Впоследствии потребуются функции проекции перехода синхронизации результирующей сети $\Sigma = (\Sigma_1 \ {}_\alpha|_\beta \ \Sigma_1)$. Эти проекции $\nu_1 : T_1 \otimes T_2 \to \mathcal{M}(T_1)$, $\nu_2 : T_1 \otimes T_2 \to \mathcal{M}(T_2)$ определяются следующим образом. Пусть $t = \mu_1 + \mu_2 \in \in T_1 \otimes T_2$, где $\mu_1 \in \mathcal{M}(T_1)$ и $\mu_2 \in \mathcal{M}(T_2)$. Тогда $\nu_1(t) = \mu_1$, $\nu_2(t) = \mu_2$. Поэтому можно писать $t = \mu_1(t) + \mu_2(t)$. Также можно расширить ν_1 и ν_2 на мультимножества переходов $\nu_i : \mathcal{M}(T_1 \otimes T_2) \to \mathcal{M}(T_i), i \in \{1, 2\}$, естественным способом: $\nu_i(\Theta) = \sum \Theta(t) \nu_i(t)$.

Далее устанавливаются соотношения в поведении исходных и результирующих сетей.

Предложение 3.4. *Пусть* $\Sigma = (\Sigma_1 \ {}_\alpha|_\beta \ \Sigma_2)$ *есть композиция сетей Петри. Пусть также* $M_1, M_1' \in \mathcal{M}(S_1)$, $M_2, M_2' \in \mathcal{M}(S_2)$.

1. *Если* $\Theta \in T_1 \otimes T_2$ *и* $\Sigma : M_1 + M_2 [\Theta\rangle M_1' + M_2'$, *то* $\Sigma_1 : M_1\ [\nu_1(\Theta)\rangle\ M_1'$ *и* $\Sigma_2 : M_2\ [\nu_2(\Theta)\rangle\ M_2'$.
2. *Если* $\Sigma_1 : M_1\ [\Theta_1\rangle M_1'$, $\Sigma_2 : M_2\ [\Theta_2\rangle M_2'$ *и* $\tau \neq \sigma_\alpha(\Theta_1) = \overline{\sigma_\beta(\Theta_2)}$, *то существует* $\Theta \in T_1 \otimes T_2 : \nu_1(\Theta) = \Theta_1, \nu_2(\Theta) = \Theta_2$.

■ 3.4

Доказательство следует из определения операции композиции объектов.

Первая часть предложения гласит, что каждому срабатыванию шага Θ в результирующей сети можно сопоставить срабатывания шагов Θ_1 и Θ_2 в исходных сетях, причем последние являются его проекциями, т.е. $\Theta_1 = \nu_1(\Theta)$ и $\Theta_2 = \nu_2(\Theta)$. Вторая часть, наоборот, утверждает, что, если в исходных сетях могут выполниться шаги Θ_1 и Θ_2 с одинаковой видимостью, т.е. $\sigma_\alpha(\Theta_1) = \overline{\sigma_\beta(\Theta_2)}$, то в результирующей сети может выполниться шаг Θ, проекциями которого являются исходные шаги Θ_1 и Θ_2.

4. ФОРМАЛЬНОЕ ОПРЕДЕЛЕНИЕ ОБЪЕКТА СЕТИ ПЕТРИ

Как было упомянуто во введении, объект сети Петри – это некоторый логический модуль с несколькими точками доступа, предназначенными для взаимодействия с другими модулями. Схематически объект сети Петри может быть изображен как прямоугольник с исходящими линиями, представляющими точки доступа. На рис. 4 (i) изображен пример схематического представления объекта. Формальное

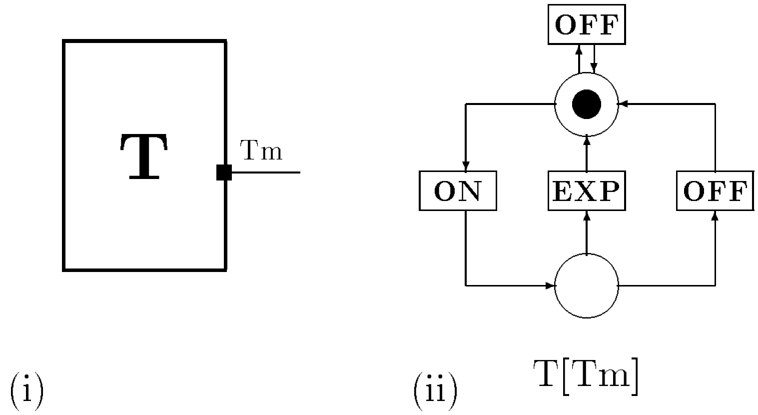

Рис. 5. Объект таймера.

определение объекта сети Петри дается следующим образом.

Определение 4.1. *Объект сети Петри.*

Объект сети Петри (СП-объект или просто объект, для краткости) есть набор $E = \langle \Sigma, \Gamma \rangle$ такой, что

1. $\Sigma = \langle S, T, {}^\bullet(), ()^\bullet, M_0 \rangle$ есть сеть Петри, называемая структурой объекта,
2. $\Gamma = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ есть множество точек доступа, каждая из которых имеет вид $\alpha_i = \langle id_i, \lambda_i \rangle$, где
   (a) id_i – имя точки доступа α_i и
   (b) $\lambda_i = \langle \Delta_i, \sigma_i \rangle$ – мультипометка сети Σ.

$\blacksquare$ 4.1

Далее СП-объекты будут обозначаться буквами E, F, G, возможно, с индексом, точки доступа – буквами α, β, γ. Для определения множества имен объекта $E = \langle \Sigma, \Gamma \rangle$ будет использоваться функция $Id(E) = \{id_a \mid \alpha \in \Gamma\}$. Часто будем писать $\alpha_i = \langle id_i, \Delta_i, \sigma_i \rangle$. То есть каждая точка доступа α_i определяется именем id_i, алфавитом имен Δ_i и пометочной функцией σ_i, сопоставляющей каждому переходу либо символ τ, либо мультимножество, определенное на множестве $\Delta_i \cup \overline{\Delta_i}$.

Определение объекта требует некоторых пояснений. СП-объект есть не что иное, как сеть Петри, снабженная множеством (мульти)пометок, т.е. объект есть обобщение помеченных сетей Петри. Взаимодействовать с объектом можно только через точки доступа. В частности, наблюдение за поведением сети является частным случаем взаимодействия. Ясно, что объект в различных точках доступа может вести себя различными способами. Переход, помеченный символом τ в определенной точке доступа, является "невидимым" в этой точке доступа, и через него нельзя взаимодействовать с объектом. Один и тот же переход может быть видим сразу из нескольких точек доступа, но, возможно, под различными именами. Более того, переход может быть видим из одной точки и невидим из другой. И, наконец, переход может вообще быть невидим из всех точек доступа.

Объект E с точками доступа $\alpha_1, \ldots, \alpha_n$ будет обозначаться как $E[\alpha_1, \ldots, \alpha_n]$. Графически объекты будут изображаться как сети Петри, в которых каждый переход помечен набором $\langle \alpha_1, \ldots, \alpha_n \rangle \in Act_1 \times \ldots \times Act_n$. Очевидно, что, с точки зрения внешнего поведения сети, выбор конкретных множеств S и T в определении сети объекта не имеет значения, и поэтому объект определяется с точностью до изоморфизма.

На рис. 4 приведен пример объекта простого протокола, содержащий схематическое и сетевое представление. Объект имеет три точки доступа. Точки Su и Sl являются точками доступа к сервису более высокого и более низкого уровней. Они определяются на алфавитах $\Delta_{Su} = \{uCon, uDis, uDat\}$ и $\Delta_{Sl} = \{lCon, lDat, lAck, lDis\}$. Точка доступа Tm предназначена для взаимодействия с объектом таймера и определена на алфавите $\Delta_{Tm} = \{ON, OFF, EXP\}$.

Другой пример объекта показан на рис. 5. Объект специфицирует механизм протокольного таймаута. Он имеет только одну точку

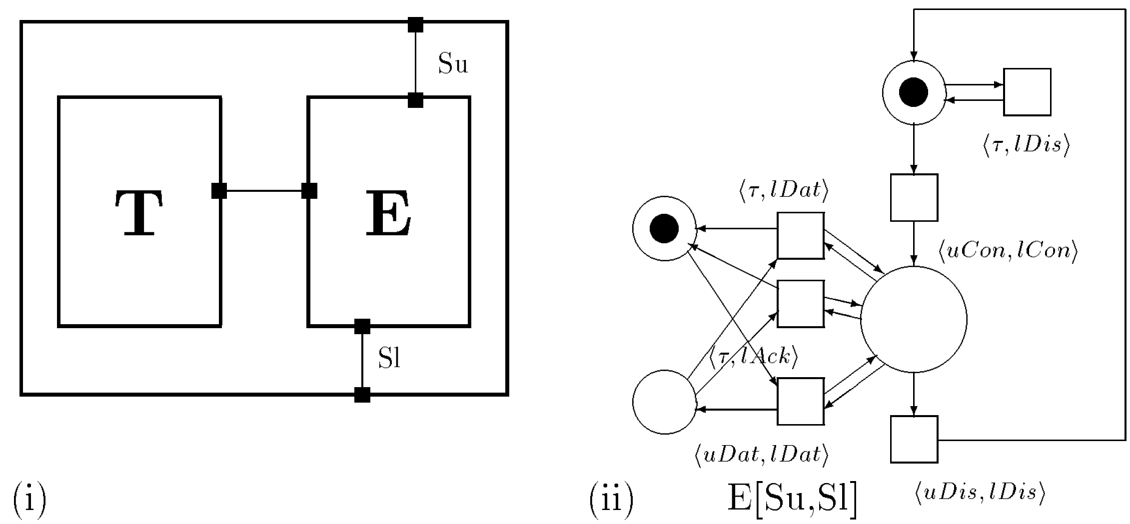

Рис. 6. Пример композиции объектов.

доступа, определенную на алфавите $\Delta_{us} = \{ON, OFF, EXP\}$, где метки соответствуют включению таймаута (ON), выключению (OFF) и его истечению (EXP).

5. ОПЕРАЦИЯ КОМПОЗИЦИИ СП-ОБЪЕКТОВ

В данном разделе вводится операция композиции объектов, позволяющая строить сложные конструкции из более простых. Предварительно дается вспомогательная процедура нормализации.

Определение 5.1. *Процедура α-нормализации.*

Пусть дан объект $E = \langle \Sigma, \Gamma$ с точками доступа $\alpha, \beta \in \Gamma\rangle$ такими, что $id_\alpha = id_\beta$. Объединение этих точек доступа есть новая точка $\gamma = \langle id_\gamma, \Delta_\gamma, \alpha_\gamma\rangle$ такая, что $id_\gamma = id_\alpha = id_\beta$, $\Delta_\gamma = \Delta_\alpha \cup \Delta_\beta$ и для всех $t \in T$

$$\sigma_\gamma(t) = \begin{cases} \sigma_\alpha(t), & \text{если } \sigma_\beta(t) = \tau; \\ \sigma_\beta(t), & \text{если } \sigma_\alpha(t) = \tau; \\ \sigma_\alpha(t) + \sigma_\beta(t), & \text{в противном случае.} \end{cases}$$

Объединение всех точек доступа с одинаковым именем называется α-нормализацией объекта и будет обозначаться как $\alpha - norm(E)$. ■ 5.1

Далее мы будем считать все объекты α-нормализованными.

Определение 5.2. *Композиция СП-объектов.*

Пусть даны объекты $E_1 = \langle \Sigma_1, \Gamma_1\rangle$, $E_2 = \langle \Sigma_2, \Gamma_2\rangle$ в нормальной форме, и $\alpha \in \Gamma_1$, $\beta \in \Gamma_2$, – их точки доступа такие, что $id_\alpha = id_\beta$. Тогда композиция E_1 и E_2 по отношению к α и β есть объект $E' = (E_1 \;{}_\alpha\|_\beta\; E_2) = \alpha - norm(E)$, где $E = \langle \Sigma, \Gamma\rangle$ и

1. $\Sigma = \partial_{\{\tilde{\alpha}, \tilde{\beta}\}}(\Sigma_1 \;{}_\alpha|_\beta\; \Sigma_2)$;
2. $\Gamma = \{\tilde{\gamma} \mid \gamma \in \Gamma_1 \cup \Gamma_1 \setminus \{\alpha, \beta\}\}$, где $\tilde{\gamma} = \langle id_\gamma, \tilde{\lambda}_\gamma\rangle$ (определение расширения пометки см. в определении 3.3).

■ 5.2

Менее формально, структура результирующего объекта строится следующим образом.

1) Выполняется композиция сетей Петри Σ_1 и Σ_2 через точки доступа α и β.

2) Над результирующей сетью выполняется операция ограничения в точках доступа α и β, участвующих в композиции (более строго – в их расширениях $\tilde{\alpha}$ и $\tilde{\beta}$). Следует заметить, что переходы, видимые из этих точек доступа, но не участвующие в композиции, также удаляются, что соответствует интуитивному смыслу пометки перехода – переход помечается исключительно для взаимодействия. Если же он помечен, но не участвует во взаимодействии – от также должен быть удален.

3) Точки доступа, не участвующие в композиции, расширяются на всю результирующую сеть и объединяются в результирующее множество точек доступа.

4) Точки доступа, участвующие в композиции (α и β), удаляются.

5) Над результирующим объектом E выполняется процедура α-нормализации, так как объединение точек доступа могло привести к появлению двух точек с одинаковым идентификатором.

На рис. 6 показан пример композиции объектов, изображенных на рис. 4 и рис. 5. Это композиция простого протокольного объекта и объекта таймера. Результирующий объект имеет только две точки доступа, соответствующие интерфейсу с верхним (Su) и нижним (Sl) уровнями. Точки доступа, участвующие в синхронизации, удалены, как сыгравшие свою роль.

В этом месте становится ясно, почему появилась необходимость в использовании мультипометки взамен обычной пометочной функции. Дело в том, что композиция объектов, где все точки доступа определены с помощью пометок, может привести к результирующему объекту, где переходу будет соответствовать несколько символов. Действительно, предположим, что при композиции двух объектов $E_1[\alpha, \zeta]$ и $E_2[\beta, \xi]$ через точки α и β сливаются переходы t_1 и t_2. Пусть $id_\zeta = id_\xi$ и $\sigma_\zeta(t_1) = \overline{a}$, $\sigma_\xi(t_2) = b$. Тогда при слиянии t_1 с t_2 результирующему переходу будет соответствовать два символа (два элементарных взаимодействия) $\overline{a}$ и b!

6. ПОНЯТИЕ ЭКВИВАЛЕНТНОСТИ ОБЪЕКТОВ

Общепризнана важность и практическая польза понятия эквивалентности в теориях параллельных и распределенных процессов. Известен широкий спектр понятий эквивалентности различной силы, определяемых в различных контекстах, см., например, обзор [30]. Большинство из этих определений могут быть выражены в терминах теории сетей Петри [31]. В данной работе будет использовано понятие бисимуляционной эквивалентности, исследованной многими авторами [2, 4, 12]. Предварительно дается базовое определение слабой бисимуляционной эквивалентности помеченных сетей Петри [32]:

Определение 6.1. *Бисимуляционная эквивалентность помеченных сетей Петри.*

Пусть даны две сети Петри Σ_1 и Σ_2, а также их пометочные функции λ_1 и λ_2, соответственно. Тогда эти сети слабо бисимуляционно эквивалентны по отношению к этим пометкам, если существует отношение бисимуляции $\Re \subseteq [M_{01}\rangle \times [M_{02}\rangle$ такое, что

1. $(M_{01}, M_{02}) \in \Re$,
2. если $(M_1, M_2) \in \Re$ и $W \in (\mathcal{M}(Vis))^*$, то
   (a) если $M_1\ (W\rangle_{\lambda_1}\ M_1'$, то $\exists M_2' : M_2\ (W\rangle_{\lambda_2}\ M_2'$ и $(M_1', M_2') \in \Re$,
   (b) если $M_2\ (W\rangle_{\lambda_2}\ M_2'$, то $\exists M_1' : M_1\ (W\rangle_{\lambda_1}\ M_1'$ и $(M_1', M_2') \in \Re$.

Эта эквивалентность будет записываться как $(\Sigma_1, \lambda_1)\ \approx^{\Re}\ (\Sigma_2, \lambda_2)$. ■ 6.1

Используя это определение как базовое, далее, однако, будем применять его версию [32], определяемую в терминах одного шага, что обеспечивается следующим фактом.

Предложение 6.2. *Одношаговая версия бисимуляционной эквивалентности.*

Пусть даны сети Петри Σ_1 и Σ_2 с пометками λ_1 и λ_2. Равенство $(\Sigma_1, \lambda_1)\ \approx^{\Re}\ (\Sigma_2, \lambda_2)$ выполняется тогда и только тогда, когда существует отношение $\Re \subseteq [M_{01}\rangle \times [M_{02}\rangle$ такое, что

1. $(M_{01}, M_{02}) \in \Re$,
2. *если* $(M_1, M_2) \in \Re$, *то*
   (a) $M_1[\Theta\rangle M_1', \Rightarrow \exists \Phi \in (\mathcal{M}(T))^* : M_2[\Phi\rangle M_2'$, $(M_1', M_2') \in \Re$ и $\sigma_\alpha(\Theta) = \sigma_\beta(\Phi)$,
   (b) *и наоборот.*

Доказательство абсолютно эквивалентно доказательству предложения 3.2 в [32]. ■ 6.2

Иными словами, если из маркировки одной сети сделать шаг Θ с видимостью $\sigma_\alpha(\Theta) = W$, то из эквивалентной маркировки другой сети можно сделать последовательность шагов Φ с той же видимостью $\sigma_\beta(\Phi) = W$. Последнее означает, что Φ можно представить в виде $\Rightarrow\Theta'\Rightarrow$, где $\sigma_\beta(\Theta') = W$. То есть выполняется последовательность, содержащая единственный видимый шаг Θ', а остальные шаги – невидимые.

Далее, основываясь на этих определениях, вводим эквивалентность СП-объектов в точках доступа.

Определение 6.3. *Эквивалентность СП-объектов в точках доступа.*

Пусть даны два объекта $E_1 = \langle\Sigma_1, \Gamma_1\rangle$ и $E_2 = \langle\Sigma_2, \Gamma_2\rangle$ с точками доступа $\alpha = \langle id_\alpha, \lambda_\alpha\rangle \in \Gamma_1$ и $\beta = \langle id_\beta, \lambda_\beta\rangle \in \Gamma_2$. Эти объекты эквивалентны в точках α и β с отношением $\Re$, если только $id_\alpha = id_\beta$ и $(\Sigma_1, \lambda_\alpha) \approx^{\Re} (\Sigma_2, \lambda_\beta)$. Эта эквивалентность записывается как $E_1\ {}_\alpha{\approx}^{\Re}_\beta\ E_2$. ■ 6.3

Полная эквивалентность, т.е. эквивалентность сразу во всех точках доступа, определяется следующим образом:

Определение 6.4. *Эквивалентность СП-объектов.*

Два объекта $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ и $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ эквивалентны, если существует отношение $\Re$ и взаимнооднозначное отображение $\omega : \Gamma_1 \to \Gamma_2$ такие, что для любого $\alpha \in \Gamma_1$ справедливо $id_\alpha = id_{\omega(\alpha)}$ и $E_1\ {}_\alpha{\approx}^{\Re}_{\omega(\alpha)}\ E_2$. Эта эквивалентность записывается как $E_1 \approx E_2$. ■ 6.4

Другими словами, эквивалентные объекты имеют одинаковое количество точек доступа с одинаковым множеством имен: $Id(E_1) = Id(E_2)$. Более того, они эквивалентны во всех точках доступа с одинаковыми именами, причем отношение бисимуляции одно и то же – $\Re$.

Нетрудно показать, что из эквивалентности объектов следует их эквивалентность в точках доступа с одинаковыми идентификаторами. Обратное, вообще говоря, не верно, то есть из эквивалентности в каждой точке не следует эквивалентность в целом. Каждая такая эквивалентность может иметь различные отношения бисимуляции, в то время как для полной эквивалентности требуется единственное отношение.

Предложение 6.5. *Пусть даны объекты $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$ $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$ с точками доступа $\alpha_1, \beta_1 \in \Gamma_1$, $\alpha_2, \beta_2 \in \Gamma_2$, $id_{\alpha_1} = id_{\alpha_2}$, $id_{\beta_1} = id_{\beta_2}$. Пусть эти объекты эквивалентны, $E_1 \approx E_2$, с отношением бисимуляции $\Re$. Если $\Sigma_1 : M_1\ [\Theta_1\rangle\ M_1'$, $\sigma_{\alpha_1}(\Theta_1) = W_\alpha$, $\sigma_{\beta_1}(\Theta_1) = W_\beta$ и $(M_1, M_2) \in \Re$, то в сети Σ_2 существует последовательность Θ_2 такая, что $\Sigma_2 : \underline{M_2 \Rightarrow} \Rightarrow \Theta_2 \Rightarrow \underline{M_2'}$, $(M_1', M_2') \in \Re$, $\sigma_{\alpha_2}(\Theta_2) = \overline{W_\alpha}$ и $\sigma_{\beta_2}(\Theta_2) = \overline{W_\beta}$.*

Доказательство. То есть необходимо доказать, что, если в одной сети выполняется шаг Θ_1 с видимостью из α_1 и β_1 как W_{α_1} и W_{β_1}, то в эквивалентной последовательности в другой сети будет содержаться единственный видимый переход Θ_2 с той же видимостью из точек α_2 и β_2. Докажем это утверждение. Пусть в эквивалентной последовательности во второй сети видимые шаги из α_2 и β_2 не совпадают. Без ограничения общности положим:

$$M_2 \Rightarrow M_2^1\ [\Theta_2'\rangle\ M_2^2 \Rightarrow M_2^3\ [\Theta_2''\rangle\ M_2^4 \Rightarrow M_2',$$

где $\sigma_{\alpha_2}(\Theta_2') = W_\alpha$, $\sigma_{\beta_2}(\Theta_2'') = W_\beta$, $\sigma_{\alpha_2}(\Theta_2'') = \tau$, $\sigma_{\beta_2}(\Theta_2') = \tau$. Однако, при единственном отношении бисимуляции $\Re$ этого быть не может, так как для, скажем, маркировки M_2^2 нет эквивалентной в первой сети. Действительно, возможно только два случая: $(M_1, M_2^2) \in \Re$ и $(M_1', M_2^2) \in \Re$. $(M_1, M_2^2) \notin \Re$, так как из β_1 видно, что W_β уже выполнился, а из β_2 – еще нет. Симметрично $(M_1', M_2^2) \notin \Re$. ■ 6.5

Известно, что одним из главных преимуществ бисимуляционной эквивалентности являются ее хорошие алгебраические свойства. Оказывается, что эквивалентность объектов наследует эти свойства.

Теорема 6.6. *Конгруэнтность эквивалентности объектов.*

Эквивалентность СП-объектов есть конгруэнтность по отношению к операции композиции, т.е. если $E_2 \approx E_3$, $\beta \in \Gamma_2$, $\gamma \in \Gamma_3$, $id_\beta = id_\gamma$, то

$$(E_1\ {}_\alpha\|_\beta\ E_2) \approx (E_1\ {}_\alpha\|_\gamma\ E_3). \tag{1}$$

Доказательство. Пусть $E_1 = \langle \Sigma_1, \Gamma_1 \rangle$, $E_2 = \langle \Sigma_2, \Gamma_2 \rangle$, $E_3 = \langle \Sigma_3, \Gamma_3 \rangle$. Обозначим $F = (E_1\ {}_\alpha\|_\beta\ E_2)$ и $G = (E_1\ {}_\alpha\|_\gamma\ E_3)$. Пусть $E_2 \approx E_3$ с отношением $\Re_1$.

Для доказательства равенства (1) достаточно показать, что выполняется отношение

$$\begin{aligned}\Re = \{&(M_1{+}M_2, M_1{+}M_3) \mid (M_2, M_3) \in \Re_1,\\ &M_1{+}M_2 \in [M_{F0}\rangle, M_1 + M_3 \in [M_{G0}\rangle\}.\end{aligned} \tag{2}$$

Это следует из предложений 6.2 и 3.4. Действительно, пусть $(M_1 + M_2, M_1 + M_3) \in \Re$, и для некоторого перехода синхронизации $t \in T_1 \otimes T_2$ выполняется

$$\Sigma_F : M_1 + M_2\ [t\rangle\ M_1' + M_2' \tag{3}$$

с $\sigma_\phi(\Theta) = W$ для некоторой $\phi \in \Gamma_2$. Покажем, что существует последовательность шагов Φ такая, что $\Sigma_G : M_1{+}M_3\ [\Phi\rangle\ M_1'{+}M_3'$ и $\sigma_\phi(\Phi) = W$. Из шага (3) и предложения 6.2 (1) следует, что в сети Σ_2 выполняется шаг $M_2\ [\nu_2(\Theta)\rangle\ M_2'$. Из эквивалентности $E_2 \approx E_3$ следует, что в Σ_3 существует последовательность $\Phi' = \underline{\ \Rightarrow \Theta' \Rightarrow\ }$ с $\sigma_\phi(\nu_2(\Theta')) = W$, $\sigma_\beta(\Theta') = \overline{W}$. Возьмем M_3 такое, что $(M_1, M_3) \in \Re_1$. Согласно предложению 6.2, $\Sigma_3 : M_3 \Rightarrow [\Theta\rangle \Rightarrow M_3'$, причем $\sigma_\gamma(\Theta_3) = \sigma_\beta(\Theta')$ и $\sigma_\phi(\Theta_3) = W$. Согласно предложению 6.2 (1), в сети G есть $t' = \Theta + \Theta_3$ с $\sigma_\phi(t') = W$. А это означает, что

$$\Sigma_G : M_1{+}M_3\ [t'\rangle\ M_1'{+}M_3' \tag{4}$$

с $\sigma_\phi(t') = W$.

Для переходов не из множества переходов синхронизации доказательство очевидно. Нетрудно обобщить этот результат и на шаг переходов. ■ 6.6

Этот результат позволяет использовать модульный подход в разработке и верификации сложных взаимодействующих систем. В частности, можно заменять один модуль в системе на эквивалентный, не изменяя поведения всей системы в целом.

Заметим, что эквивалентность объектов, построенная на шаговой семантике, есть наименее сильная эквивалентность, гарантирующая ее конгруэнтность по отношению к композиции. Нетрудно убедиться, что эквивалентность, основанная на интерливинговой семантике сетей Петри, не обладает свойством конгруэнтности.

7. СВОЙСТВА ОПЕРАЦИИ КОМПОЗИЦИИ ОБЪЕКТОВ

Для работы с композицией объектов на практике необходимо, чтобы операция обладала свойствами коммутативности и ассоциативности.

Предложение 7.1. *Коммутативность композиции объектов.*

Пусть даны объекты E, F и их точки доступа $\alpha \in \Gamma_E$, $\beta \in \Gamma_F$. Тогда $(E\ {}_\alpha\|_\beta\ F) = (F\ {}_\beta\|_\alpha\ E)$.

Доказательство немедленно следует из коммутативности операции объединения (для множеств и мультимножеств) и определения 5.2. ■ 7.1

Оказывается также, что операция композиции ассоциативна:

Теорема 7.2. *Ассоциативность операции композиции объектов.*

Пусть E_1, E_2, E_3 есть объекты, $\alpha \in \Gamma_1$, $\beta, \zeta \in \Gamma_2$, $\xi \in \Gamma_3$, $id_\alpha \notin Id(E_3)$ и $id_\xi \notin Id(E_1)$. Тогда

$$((E_1\ {}_\alpha\|_\beta\ E_2)\ {}_\zeta\|_\xi\ E_3) = (E_1\ {}_\alpha\|_\beta\ (E_2\ {}_\zeta\|_\xi\ E_3)).$$

Доказательство. Обозначим $F = ((E_1\ {}_\alpha\|_\beta\ E_2)\ {}_\zeta\|_\xi\ E_3)$, $G = (E_1\ {}_\alpha\|_\beta\ (E_2\ {}_\zeta\|_\xi\ E_3))$. Очевидно, достаточно проверить, что множества переходов синхронизации сетей Σ_F и Σ_F равны, т.е. что $(T_1 \otimes T_2) \otimes T_3 = T_1 \otimes (T_2 \otimes T_3)$. Для этого продолжим эксплуатацию идеи доказательства теоремы 3.2. Покажем, что оба множества равны $T_1 \otimes T_2 \otimes T_3$, которое определяется следующим образом. Строится вспомогательная сеть $\tilde\Sigma_{123} = \langle S_{12} \cup S_{23}, T_1 \cup T_2 \cup T_3, {}^\bullet(), ()^\bullet, M_0\rangle$, где $S_{12} = \Delta_{12} = \Delta_\alpha \cup \Delta_\beta$, $S_{23} = \Delta_{23} = \Delta_\zeta \cup \Delta_\xi$, $M_0 = \mathbf{0}$,

$$^\bullet(t) = \begin{cases} \overline{\sigma_\alpha^+(t)}\lceil\Delta_{12}, & \text{если } t \in T_1; \\ \overline{\sigma_\beta^+(t)}\lceil\Delta_{12} + \overline{\sigma_\zeta^+(t)}\lceil\Delta_{23}, & \text{если } t \in T_2; \\ \overline{\sigma_\xi^+(t)}\lceil\Delta_{23}, & \text{если } t \in T_3; \end{cases}$$

$$(t)^\bullet = \begin{cases} \sigma_\alpha^+(t)\lceil\Delta_{12}, & \text{если } t \in T_1; \\ \sigma_\beta^+(t)\lceil\Delta_{12} + \sigma_\zeta^+(t)\lceil\Delta_{23}, & \text{если } t \in T_2; \\ \sigma_\xi^+(t)\lceil\Delta_{23}, & \text{если } t \in T_3. \end{cases}$$

Множество минимальных Т-инвариантов для $\tilde\Sigma_{123}$ дает множество $T_1 \otimes T_2 \otimes T_3$. Рассмотрим, как вычисляется множество $(T_1 \otimes T_2) \otimes T_3$. Сначала, используя сеть $\tilde\Sigma_{12}$, вычисляется множество $T_1 \otimes T_2 = \{x_1, \ldots, x_p\}$, где $x_i = \sum_{t \in T_1} n_t t + \sum_{t \in T_2} n_t t$ минимальны. Затем строится сеть $\tilde\Sigma_{12,3}$ из переходов $T_1 \otimes T_2$ и T_3, предполагая, что

$$\begin{aligned}(x_i)^\bullet &= \sigma_\zeta(\textstyle\sum_{t \in T_1} n_t t + \sum_{t \in T_2} n_t t) = \\ &= \sigma_\zeta(\textstyle\sum_{t \in T_2} n_t t) = \sum_{t \in T_2} n_t \sigma_\zeta(t).\end{aligned}$$

Известно, что алгоритмы поиска Т-инвариантов (см., например, [28]) заключаются в пошаговой трансформации матрицы инцидентности путем линейной композиции других строк для обнуления столбцов. Каждый такой шаг может быть проинтерпретирован как редукция сети Петри, изолирующая место, соответствующее обнуляемому столбцу (см. правило редукции R1 в [33]). Будем выполнять алгоритм нахождения Т-инвариантов в сети $\tilde\Sigma_{123}$ так, чтобы на первых шагах обнулились столбцы, соотвествующие местам S_{12}. Строки полученной матрицы инцидентности будут соответствовать переходам из $T_1 \otimes T_2$ и T_3. Заметим, что эта матрица соответствует сети $\tilde\Sigma_{12,3}$. Следовательно, продолжение применения алгоритма приведет к множеству инвариантов, одновременно соответствующих сетям $\tilde\Sigma_{123}$ и $\tilde\Sigma_{12,3}$. Аналогично показывается, что $T_1 \otimes (T_2 \otimes T_3) = T_1 \otimes T_2 \otimes T_3$. То есть выполняется равенство $T_1 \otimes (T_2 \otimes T_3) = (T_1 \otimes T_2) \otimes T_3$. ■ 7.2

Этот результат позволяет писать $(E_1\ {}_\alpha\|_\beta\ E_2\ {}_\zeta\|_\xi\ E_3)$ и выполнять композиции сетей, не заботясь о порядке применения операции.

8. ЗАКЛЮЧЕНИЕ

В данной работе введено понятие объекта сети Петри и операция их композиции. Объект сети Петри является обобщением помеченных сетей Петри, где вместо одной пометки используется несколько пометок, которые называются точками доступа к объекту. Такое обобщение позволяет сопоставлять выполнению одного физического события (срабатывание перехода) выполнение нескольких логических событий по взаимодействию с другими объектами. Это, в свою очередь, потребовало обобщения понятия пометки до мультипометки, когда одному переходу сопоставляется мультимножество символов.

Введение операции композиции обеспечивает композиционный стиль разработки и верификации систем. Действительно, он позволяет разрабатывать системы по частям с последующей их композицией. Более того, свойство конгруэнтности операции композиции позволяет заменять подсистемы на эквивалентные, не изменяя поведения системы в целом. Это может быть полезным при проектировании путем пошаговой детализации. В работах [23, 25] приведен пример применения этого принципа при иерархической композиции протоколов.

На наш взгляд, достаточно важное значение для практики имеет возможность схематического представления структур объектов сетей Петри. Действительно, разрабатывая систему в стиле "сверху-вниз", разработчик сначала конструирует систему в виде структуры объектов в схематической форме, что соответствует привычной практике проектирования в терминах блоков и соединений. Допускается любая степень вложенности блоков. На последующем этапе для наиболее внутренних блоков определяется их внутренняя структура в терминах сетей Петри. Заметим, что на всех этапах такое проектирование является формальным. В работах [25, 34, 35] приводятся некоторые результаты применения подхода к разработке реальных параллельных и распределенных систем.

Можно наметить дальнейшие проблемы, решение которых приблизит применение композициональных сетей Петри к решению практических задач.

а) Для конструирования внутренней структуры объектов необходим более развитый набор операций над сетями Петри. Это могут быть операции последовательной композиции, итерации, выбора, и др. В этом плане может быть разумным расширить понятие точки доступа на места сети Петри.

б) Для дальнейшего усиления практических свойств формализма необходимо провести работу по его расширению и обобщению на сети Петри высокого уровня, установлению взаимоотношения с параллельными языками программирования и спецификации, разработке соответствующих средств автоматизации.

СПИСОК ЛИТЕРАТУРЫ

1. *Olderog E.-R.* Operational Petri Net Semantics for CCSP // Lecture Notes in Computer Sciences. V. 266. Springer-Verlag, 1984.

2. *Milner R.* A Calculus for Communication Systems // Lecture Notes in Computer Sciences. V. 92. Springer-Verlag, 1980.

3. *Хоар Ч.* Взаимодействующие последовательные процессы. М: Мир, 1989.

4. *Bergstra J.A., Klop J.W.* Algebra of Communicating Processes / Math. on Comput. Sci. Proc. CWI Symp. 1986. P. 89–138.

5. *Bolognesi T., Brinksma E.* Introduction to the ISO Specification Language LOTOS // Computer Networks and ISDN Systems. 1987. V. 14. P. 25–89.

6. *Hopkins R., Hall J., Botti O.* A Basic-net Algebra for Programs Semantics and Application to OCCAM // Lecture Notes in Computer Scences. V. 609. Springer-Verlag, 1992. P. 179–214.

7. *Mauw S., Veltink G.J.* A Process Specification Formalism // Fundamenta Informaticae. 1990. V. XIII. P. 85–139.

8. *Котов В.Е.* Сети Петри. М.: Наука, 1984.

9. Reisig W., Rozenberg G. (eds). Lecture Notes on Petri Nets. Parts I and II // Lecture Notes in Computer Sciences. V. 1491–1492. Springer-Verlag, 1998.

10. *Goltz U.* On Representing CCS Programs by Finite Petri Nets // Lecture Notes in Computer Sciences. V. 324. Springer-Verlag, 1988. P. 339–350.

11. *Taubner D.* Finite Representation of CCS and TCSP Programs by Automata and Petri Nets // Lecture Notes in Computer Sciences. V. 369. Springer-Verlag, 1989.

12. *Van Glabbeek R.J., Vaandrager F.W.* Petri Net Models for Algebraic Theories of Concurrency // Lecture Notes in Computer Sciences. V. 259. Springer-Verlag, 1987. P. 224–242.

13. *Barbeau M., Bochmann G.V.* A Subset of Lotos with Computational Power of Place/Transition Nets // Lecture Notes in Computer Science. 1993. V. 691. P. 49–68.

14. *Bernadinello L., De Cindio F.* A Survey of Basic Net Models and Modular Classes // Lecture Notes in Computer Sciences. V. 609. Springer-Verlag, 1992. P. 304–351.

15. *Cherkasova L.A., Kotov V.E.* Structured Nets // Lecture Notes in Computer Sciences. V. 118. Springer-Verlag, 1981. P. 242–251.

16. *Kotov V.E.* An Algebra for Parallelism Based on Petri nets. MFCS'78 // Lecture Notes in Computer Sciences. V. 64. Springer-Verlag, 1978. P. 39–55.

17. *Valette R.* Analysis of Petri Nets by Stepwise Refinements // Journal of Computer and System Science. 1979. V. 18. P. 35–46.

18. *Baumgarten B., Ochsenschläger P., Prinoth R.* Building Blocks for Distributed System Design / Diaz M. (ed.) Protocol Specification, Testing, and Verification. V. Proc. of the V IFIP WG 6.1 Conf. Elsevier Science Publishers B.V., North-Holland, 1986. P. 19–38.

19. *Vogler W.* Failures Semantics and Deadlocking of Modular Petri Nets // Acta Informatica. 1989. V. 26. P. 333–348.

20. *Voss K.* System Specification with Labelled Nets and the Notion of Interface Equivalence. Arbeitspapiere der GMD. № 221. 1986.

21. Rozenberg G. (ed.) Advances in Petri Nets 1992 // Lecture Notes in Computer Sciences. V. 609. Springer-Verlag, 1992.

22. *Best E., Devillers R., Hall J.G.* The Box Calculus: a New Causal Algebra with Milti-label Communication // Lecture Notes in Computer Sciences. V. 609. Springer-Verlag, 1992. P. 21–69.

23. *Анисимов Н.А.* Иерархическая композиция протоколов // Автоматика и вычислительная техника. 1990. № 1. С. 3–10.

24. *Anisimov N.A.* A Petri Net Entity as a Formal Model for LOTOS, a Specification Language for Distributed and Concurrent Systems / Mirenkov N.N. (ed.) Parallel Computing Technologies. Singapore: World Scientific, 1991. P. 440–450.

25. *Anisimov N.A., Koutny M.* On Compositionality and Petri Nets in Protocol Engineering / Dembiński P., Średniawa M. (eds.) Protocol Specification, Testing and Verification. V. XV. Chapman & Hall, 1996. P. 71–86.

26. Мизин И.А., Кулешов А.П. (ред.) Протоколы информационно-вычислительных сетей. Справочник. М.: Радио и связь, 1990.

27. Jensen K., Rozenberg G. (eds.) High-Level Petri Nets. Theory and Application. Springer-Verlag, 1991.

28. *Ачасова С.М., Бандман О.Л.* Корректность параллельных вычислительных процессов. Новосибирск: Наука, 1990.

29. *Alaiwan H., Toudic J.M.* Recherche de semiflots, des verrous et trappes dans les r'eseaux de Petri // T.S.I.-Technique et Sciences Informatiques. 1985. V. 4. P. 103–112.

30. *Van Glabbeek R.J.* The Linear Time – Branching Time Spectrum // Lecture Notes in Computer Sciences. V. 458. Berlin, Heidelberg, New York: Springer, 1990. P. 278–297.

31. *Pomello L., Rozenberg G., Simone C.* A Survey of Equivalence Notions for Net Based Systems // Lecture Notes in Computer Sciences. V. 609. Springer-Verlag, 1992.

32. *Best E., Devillers R., Kiehn A., Pomello L.* Concurrent Bisimulations in Petri Nets. Acta Informatica // 1991. V. 28. P. 231–261.

33. *Berthelot G., Roucairol G., Valk R.* Reduction of Nets and Parallel Programs // Lecture Notes in Computer Science. 1980. V. 84. P. 277–290.

34. *Anisimov N.A., Kovalenko A.A., Tarasov G.V., Inzartsev A.V., Sherbatyuk A.Ph.* A Graphical Environment for AUV Mission Programming and Verification. Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology. New Hermpshire, USA. September 7–10, 1997. P. 394–405.

35. *Anisimov N.A, Kovalenko A.A., Postupalski P.A., Vuong S.T.* Application of Compositional Petri Nets and PN^3-Tool to the Specification of Distributed Multimedia Objects / Chang S.K. et al. (eds.) Advances in Distributed Multimedia Systems. Singapore: World Scientific, 1999. P. 99–116.