

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 519.715

В. Н. Дубинин

АСИНХРОННОЕ МОДЕЛИРОВАНИЕ NCES-СЕТЕЙ¹

Аннотация. Рассматриваются вопросы асинхронного моделирования NCES-сетей с помощью формализма, основанного на сетях Петри. Приводятся правила трансформации NCES-сетей в асинхронную модель. Предложенный метод демонстрируется на примере. Асинхронное моделирование рассматривается как шаг к формальной верификации NCES-сетей с помощью метода *Model Checking*.

Ключевые слова: асинхронное моделирование, сетевые системы «условие–событие», сети Петри, проверка моделей, трансформация моделей.

Abstract. In the paper questions of asynchronous modelling of net condition/event systems (NCES) using a formalism based on Petri nets, are considered. Rules for transforming NCES to the asynchronous model are represented. The suggested method is illustrated by an example. The asynchronous modelling is considered as the starting point to formal verification of NCES by means of the model checking method.

Keywords: asynchronous modelling, net condition/event systems, Petri nets, model checking, model transformation.

Введение

В настоящее время для моделирования дискретно-событийных систем большое распространение получили сетевые системы «условие–событие» (NCES-сети) [1]. Как отмечено в работе [2], данные модели обладают мощностью машин Тьюринга. Одной из основных особенностей NCES-сетей является их синхронно-асинхронный характер. В работе [2] были предложены методы анализа данного вида сетевых моделей. Среди инструментальных средств анализа NCES-сетей можно отметить системы *INA*, *Sesa* [2], *Vive* [3]. По мере расширения сфер использования NCES-сетей возникает необходимость их модификации для увеличения выразительных возможностей. Например, в работе [4] были предложены арифметические NCES-сети (*aNCES*-сети) для моделирования систем функциональных блоков нового международного стандарта IEC 61499. Расширение классических NCES-сетей во многих случаях приводит к невозможности использования разработанных ранее методов и средств для их исследования.

¹ Работа выполнена в рамках аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009–2010 годы)», № гос. регистрации НИР 01200952061.

Результативным методом формальной верификации моделей с конечным, возможно очень большим, числом состояний (к каким относятся при определенных ограничениях и NCES-сети) является *Model Checking* (проверка моделей) [5]. Данный метод позволяет проектировщику определить, удовлетворяет ли модель системы определенным требуемым свойствам. Как правило, в этой технике верификации используется алгоритм проверки формул временных логик *LTL* и *CTL* [6]. Инструментальные средства, поддерживающие *Model Checking*, позволяют анализировать системы с большим числом состояний благодаря сжатому представлению пространства состояний, основанному на упорядоченных двоичных решающих диаграммах (*OBDD*). Проверка моделей нашла широкое распространение как в проектировании аппаратуры, так и программного обеспечения [6]. Она успешно использовалась в качестве метода формальной верификации недетерминированных автоматов [7], сетей Петри [8], а также сетевых моделей, основанных на сетях Петри, таких как *PRES*-сети [9] и сети Петри с дуальными переходами [10]. Наиболее известными инструментальными средствами на основе *Model Checking* являются свободно доступные системы *SMV* [5] и *NuSMV* [11].

В данной работе предлагается асинхронное моделирование как промежуточный шаг к формальной верификации NCES-сетей различных классов с использованием *Model Checking*. Моделирование NCES-сетей «в лоб» как синхронно-асинхронной системы в верификаторах типа *SMV* затруднительно из-за сложности явного вычисления разрешенных шагов [12], вычисление же разрешенных шагов в асинхронной модели производится автоматически. Следует также заметить, что, кроме верификации, асинхронная модель NCES-сетей может быть использована при реализации NCES-сетей в мультипрограммных средах и на различных аппаратных платформах.

В данной работе определяется формализм для асинхронного моделирования NCES-сетей, основанный на сетях Петри. Асинхронность означает, что срабатывания переходов не привязаны ни к каким внешним событиям и не существует механизма глобального времени для срабатывания переходов. Переход от NCES-сети к асинхронной модели (А-модели) описывается с помощью правил трансформации сетевых моделей. В заключении работы приводится демонстрационный пример.

1 NCES-сети

Ниже (без потери общности) рассматриваются замкнутые одноуровневые NCES-сети, расширенные ингибиторными дугами. Одноуровневую NCES-сеть можно получить из многоуровневой модульной NCES-сети при помощи операции компоновки.

Замкнутая одноуровневая NCES-сеть N' с ингибиторными дугами определяется следующим кортежем:

$$N' = (P, T, F, C, I, E, W_F, W_C, em, m_0),$$

где $P = \{p_1, \dots, p_n\}$ – конечное множество позиций; $T = \{t_1, \dots, t_m\}$ – конечное множество переходов; $F \subseteq (P \times T) \cup (T \times P)$ – множество обычных дуг; $C \subseteq P \times T$ – множество условных дуг; $I \subseteq P \times T$ – множество ингибиторных дуг; $E \subseteq T \times T$ – множество событийных дуг; $W_F: F \rightarrow N^+ = \{1, 2, \dots\}$ – функция весов обычных дуг; $W_C: C \rightarrow N^+$ – функция весов условных дуг; $em: T \rightarrow \{AND, OR\}$ – со-

бытийный режим работы переходов; $m_0: P \rightarrow N_0 = \{0, 1, \dots\}$ – начальная маркировка. Отношение E не содержит циклов. Маркировка m сети $NCES$ есть отображение $m: P \rightarrow N_0$. Событийный режим $em(t)$ перехода t определяет, как связаны входящие событийные дуги – конъюнктивно (AND) или дизъюнктивно (OR). По умолчанию предполагается конъюнктивная связь. Определим следующие множества для перехода $t \in T$: $ppre^t$ и $ppost^t$ – множества входных и выходных позиций перехода t , соответственно; $tpre^t$ – множество входных переходов перехода t .

Два перехода конфликтны, если срабатывание одного из них приводит к деактивации другого. Переход $t \in T$ разрешен со стороны позиций при маркировке m , если $\forall p \in ppre^t [(p, t) \in F \rightarrow m(p) \geq W_F(p, t) \ \& \ (p, t) \in C \rightarrow m(p) \geq W_C(p, t) \ \& \ (p, t) \in I \rightarrow m(p) = 0]$.

Переход $t \in T$, не имеющий входящих событийных дуг ($tpre^t = \emptyset$), называется *триггерным*. Множество триггерных переходов определяется как $T_t = \{t \in T \mid tpre^t = \emptyset\}$.

Непустое множество переходов $\xi \subseteq T$ есть шаг при маркировке m , если:

- 1) $|\mu \cap T_t| = 1$;
- 2) для каждого перехода $t \in \xi$ с $t \notin T_t$ имеет место: $em(t) = OR \rightarrow \exists t' \in \in tpre^t[t' \in \mu] \ \& \ em(t) = AND \rightarrow \forall t' \in tpre^t[t' \in \mu]$;
- 3) все переходы неконфликтны по отношению друг к другу при маркировке m . Обозначим Ξ – множество всех шагов в N .

Шаг ξ называется *разрешенным* при маркировке m , если каждый переход из ξ является разрешенным со стороны позиций при маркировке m и не существует множества переходов $\xi' = \xi \cup \{t\}$, которое также является шагом и каждый переход которого разрешен со стороны позиций при маркировке m .

Срабатывание разрешенного шага состоит из срабатывания каждого перехода, входящего в шаг. При срабатывании перехода $t \in \xi$ выполняются следующие действия:

- 1) удаляются метки из входных позиций t ;
- 2) добавляются метки в выходные позиции t . При срабатывании перехода t при маркировке m она меняется на маркировку m' в соответствии со следующим правилом:

$$m'(p) = \begin{cases} m(p) - W_F(p, t), & \text{если } (p, t) \in F \ \& \ (t, p) \notin F, \\ m(p) + W_F(t, p), & \text{если } (t, p) \in F \ \& \ (p, t) \notin F, \\ m(p) - W_F(p, t) + W_F(t, p), & \text{если } (p, t) \in F \ \& \ (t, p) \in F, \\ m(p), & \text{иначе.} \end{cases}$$

Маркировка m'' сети N является последователем маркировки m_0 под воздействием w (записывается как $m_0[w]m''$, где $w = \xi_1, \dots, \xi_n \in \Xi$), если выполняется $|w| = 0 \ \& \ m_0 = m''$ или существует маркировка m' такая, что $m_0[(\xi_1, \dots, \xi_{n-1})]m' \ \& \ m'[\xi_n]m''$. Тогда w называется переключательной последовательностью при m_0 и записывается $m_0[w]$. Определим $W_N := \{w \in \Xi \mid m_0[w] \}$ – множество всех видов переключательных последовательностей в сети N . Множество достижимых маркировок в сети N из начальной маркировки определяется как $[m_0]_N := \{m \mid \exists w \in W_N : m_0[w]m\}$.

2 Формализм асинхронной модели NCES-сетей

Асинхронная модель (в дальнейшем просто А-модель) NCES-сетей как модельная форма определяется следующим кортежем:

$$N'' = (P, T, X, Y, Z, R, U, W_X, W_Y, W_Z, m_0),$$

где P – конечное множество позиций А-модели; T – конечное множество переходов А-модели; $X \subseteq P \times T$ – множество входных дуг переходов с порогом-минимумом (дуги с проверкой на «больше»); $Y \subseteq T \times P$ – множество выходных дуг переходов; $Z \subseteq P \times T$ – множество входных дуг переходов с порогом-максимумом (дуги с проверкой на «меньше»); $R \subseteq P \times T$ – множество дуг чтения маркировок позиций; $U \subseteq (X \cup R) \times Y$ – отношение сопряженности дуг. Для отношения U должны выполняться следующие ограничения: $\forall u \in U [pr_2(pr_1(u)) = pr_1(pr_2(u))]$ и $\neg \exists u_1, u_2 \in U [pr_2(u_1) = pr_2(u_2)]$. Здесь под $pr_i(u)$ понимается i -й компонент (проекция) кортежа u . Первое ограничение говорит о том, что связанные отношением дуги должны быть инцидентны одному и тому же переходу. Второе ограничение свидетельствует о том, что входная позиция перехода может быть связана посредством сопряженных дуг только с одной выходной позицией этого перехода; $W_X: X \rightarrow N_0 \times \{N_0 \cup all\}$ – функция весов дуг типа X ; $W_Y: Y \rightarrow \{N^+ \cup *\} \times \{add, reset\}$ – функция весов дуг типа Y ; $W_Z: Z \rightarrow N^+ \times \{N_0 \cup all\}$ – функция весов дуг типа Z ; m_0 – начальная маркировка.

Первый компонент веса дуги типа X указывает минимальный порог для срабатывания перехода, а второй – число удаляемых меток из инцидентной дуге позиции. Если второй компонент равен all , то из позиции удаляются все метки. В отношении дуги типа Y : первый компонент веса дуги в общем случае определяет число меток, добавляемых в инцидентную дуге позицию, в то время как второй компонент указывает режим добавления – с предварительным обнулением маркировки этой позиции (слово *reset*) или нет (слово *add*). Если первый компонент равен $*$, то число добавляемых в выходную позицию перехода меток вычисляется и зависит от маркировок определенных входных позиций этого перехода. Первый компонент веса дуги типа Z определяет максимальный порог для срабатывания перехода, а второй – число удаляемых меток из инцидентной дуге позиции.

Переход $t \in T$ разрешен при маркировке m , если

$$\forall p \in P [(p, t) \in X \rightarrow m(p) \geq pr_1(W_X(p, t)) \ \& \ (p, t) \in Z \rightarrow m(p) < pr_1(W_Z(p, t))].$$

Введем функцию $W'_Y: Y \rightarrow N_0$, определяющую реальный вес дуги типа Y . Данную функцию определим следующим образом:

$$W'_Y(t, p) = \begin{cases} pr_1(W_Y(t, p)), & \text{если } pr_1(W_Y(t, p)) \in N^+, \\ \sum_{p \in S_1} m(p) + \sum_{x \in S_2} pr_2(W_X(x)), & \text{иначе,} \end{cases}$$

где $S_1 = \{p' \mid ((p', t), (t, p)) \in U \ \& \ ((p', t) \in X \rightarrow pr_2(W_X(p', t)) = all)\}$ и $S_2 = \{(p', t) \mid ((p', t), (t, p)) \in U \ \& \ (p', t) \in X \ \& \ pr_2(W_X(p', t)) \in N_0\}$.

При срабатывании перехода $t \in T$ при маркировке m последняя изменяется на маркировку m'' (обозначается $m \xrightarrow{t} m''$ или $m[t > m'']$) в соответствии со следующим:

$$m'(p) = \begin{cases} m(p) - pr_2(W_X(p, t)), & \text{если } (p, t) \in X \text{ \& } pr_2(W_X(p, t)) \in N^+, \\ 0, & \text{если } (p, t) \in X \text{ \& } pr_2(W_X(p, t)) = all, \\ m(p), & \text{иначе;} \end{cases}$$

$$m''(p) = \begin{cases} m'(p) + W'_Y(t, p), & \text{если } (t, p) \in Y \text{ \& } pr_2(W_Y(t, p)) = add, \\ W'_Y(t, p), & \text{если } (t, p) \in Y \text{ \& } pr_2(W_Y(t, p)) = reset, \\ m'(p), & \text{иначе.} \end{cases}$$

3 Подход к моделированию NCES-сетей с использованием А-модели

Основными моментами при моделировании NCES-сетей являются:

1) вычисление разрешенных шагов;

2) изменение маркировки позиций при срабатывании переходов шага, причем это действие должно быть выполнено как неделимое.

Как показано в работе [12], вычисление разрешенных шагов является нетривиальной задачей. В основу предлагаемого асинхронного моделирования шага положена концепция распространения событийного сигнала, источником которого является триггерный переход, через форсируемые переходы событийного графа этого триггерного перехода. Здесь под событийным графом понимается событийный граф максимально разрешенных шагов (СГМРШ) [12]. Если принятый форсируемым переходом сигнал может привести к срабатыванию этого перехода, то он (переход) присоединяется к текущему формируемому шагу, и сигнал передается дальше всем переходам, связанным с данным переходом событийными дугами. Если сработал переход, который был в конфликте с другим переходом, то второй переход деактивируется.

Утверждение 1. Свободный язык, порождаемый фрагментом А-модели NCES-сети, соответствующим событийному графу некоторого триггерного перехода NCES-сети, определяет все возможные разрешенные шаги, инициируемые этим триггерным переходом в NCES-сети.

Исходя из данного утверждения, можно заключить, что разрешенные шаги получаются в А-модели *автоматически*, как, собственно, и сама маркировка при срабатывании переходов этих шагов.

Для обеспечения неделимости действий по изменению маркировки сетевой модели вводится специальная семафорная позиция, которая сбрасывается при срабатывании триггерного перехода и устанавливается, когда срабатывание переходов, входящих в соответствующий шаг, завершено. Это определяется по отсутствию сигналов между переходами шага. Для мониторинга сигналов вводится специальный тип перехода *monitor*.

Можно выделить два подхода к реализации мониторинга событийных сигналов. В первом случае (режим 1) для каждого событийного графа вводится собственный переход-монитор. Во втором случае (режим 2) используется только один переход-монитор. Оба подхода во многом сходны, каждый из них имеет определенные преимущества и недостатки, поэтому при выборе подхода следует учитывать конкретную область применения. Второй подход удобен, например, при моделировании модульных NCES-сетей. В этом случае

бывает затруднительно вычислить событийный граф, поскольку он может быть разнесен по нескольким модулям.

Ниже кратко рассматриваются элементы А-модели и их связь с элементами NCES-сетей. Множество позиций P А-модели NCES-сети можно разбить на несколько подмножеств: $P = P_{NCES} \cup P_e \cup P_{before} \cup P_{after} \cup P_{lim} \cup P_{sem}$, причем $P_{NCES} \cap P_e \cap P_{before} \cap P_{after} \cap P_{lim} \cap P_{sem} = \emptyset$, где P_{NCES} – множество позиций исходной NCES-сети; P_e – множество событийных (сигнальных) позиций, обозначаемых в дальнейшем в виде шестиугольников, в отличие от других типов позиций, обозначаемых окружностями; P_{before} – множество позиций, являющихся двойниками некоторых позиций из множества P_{NCES} , предназначенных для представления (сохранения) их маркировки непосредственно до срабатывания шага; P_{after} – множество позиций, являющихся двойниками некоторых позиций из множества P_{NCES} , предназначенных для хранения изменений маркировки позиции-оригинала во время срабатывания шага; P_{lim} – множество позиций-ограничителей числа срабатываний переходов в шаге; P_{sem} – множество семафорных позиций. Множество семафорных позиций зависит от метода моделирования. Но при любом методе используется семафорная позиция mt , разрешающая срабатывание только одного триггерного перехода. В режиме 1 используются позиции-признаки, отмечающие, какой именно триггерный переход сработал. Имя позиции из P_{NCES} в дальнейшем будет определяться именем соответствующей позиции исходной NCES-сети. Имя позиции из P_e образуется из символа e , за которым следует числовой индекс. Имена позиций из множества P_{before} (P_{after}) образуются путем добавления знака подчеркивания после имени (перед именем) позиции-оригинала.

Множество позиций T А-модели NCES-сети разбивается на следующие подмножества: $T = T_{NCES} \cup T_{monitor} \cup T_{reset}$, причем $T_{NCES} \cap T_{monitor} \cap T_{reset} = \emptyset$, где T_{NCES} – множество переходов исходной NCES-сети; $T_{monitor}$ – множество переходов, осуществляющих мониторинг процесса срабатывания шага и срабатывающих при завершении вычисления шага; T_{reset} – множество переходов сброса маркировки событийных позиций.

Множество дуг Q А-модели NCES-сети можно разбить следующим образом: $Q = Q_F \cup Q_C \cup Q_I \cup X \cup Y \cup Z$, причем $Q_F \cap Q_C \cap Q_I \cap X \cap Y \cap Z = \emptyset$, где Q_F , Q_C и Q_I – множества дуг А-модели, соответствующих дугам типов F , C и I сети NCES. Дуги NCES-сети (за исключением событийных дуг) могут быть напрямую выражены с помощью дуг А-модели. В табл. 1 приведены семантически эквивалентные дуги NCES-сети и А-модели.

Таблица 1

Дуга NCES	Вес дуги NCES	Дуга А-модели	Вес дуги А-модели
$q \in C$	$W_C(q)$	$q' \in X$	$W_X(q') = (W_C(q), 0)$
$q \in F \cap P \times T$	$W_F(q)$	$q' \in X$	$W_X(q') = (W_F(q), W_F(q))$
$q \in F \cap T \times P$	$W_F(q)$	$q' \in Y$	$W_Y(q') = (W_F(q), add)$
$q \in I$	–	$q' \in Z$	$W_Z(q') = (1, 0)$

В дальнейшем для «облегчения» рисунков будут использоваться следующие обозначения для дуг А-модели. Дугу типа Y с весом $(*, all)$ будем отмечать символом «+», а дугу с весом $(*, reset)$ – символом «*», дугу чтения – пунктирной линией. Для обозначения дуги с проверкой на «меньше» исполь-

зуется нотация ингибиторных дуг, но при этом рядом с дугой указывается ее вес. Если вес обычной дуги не указан, то он полагается равным (1,1). Дугу типа X с весом (0, *all*) будем называть дугой сброса. Сопряженные дуги графически будем отображать таким образом, что оконечная вершина первой дуги будет начальной вершиной второй дуги.

4 Правила преобразования NCES в А-модель

В общем случае процесс преобразования NCES-сетей в А-модели можно представить в виде последовательности графовых трансформаций [13]. С целью упрощения в правилах трансформации графов будет представлена только наиболее важная (в точки зрения существа проблемы) информация.

При использовании режима 1 должны быть выполнены предварительные вычисления, в частности, должны быть вычислены все событийные графы. При использовании режима 2 этого делать не требуется. На начальной фазе используются правила создания семафорных позиций и переходов-мониторов. При использовании режима 1 создается позиция-признак и переход-монитор для каждого событийного графа. При использовании режима 2 создается только один переход-монитор, а позиции-признаки не используются. Для обоих режимов создается семафорная позиция *mt*.

Для каждой пары форсируемых переходов, связанных событийной дугой, генерируется событийная позиция, связанная с этими переходами (рис. 1).

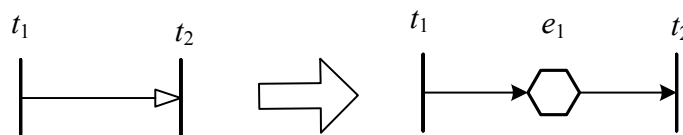


Рис. 1 Правило создания событийной позиции

Каждому форсируемому ИЛИ-переходу NCES-сети, имеющему более двух входящих событийных дуг, ставится в соответствие одна позиция-ограничитель и дуга, связывающая эту позицию с данным переходом. При этом триггерный переход – источник шага – также связывается дугой с этой позицией для ее установки. Кроме того, вводится переход для сброса входной событийной позиции перехода при нулевой маркировке позиции-ограничителя и соответствующие дуги – ингибиторная и обычная. Схематично данное правило проиллюстрировано на рис. 2. Штриховой контур акцентирует внимание на том, что преобразование осуществляется внутри событийного графа триггерного перехода t_i .

Необходимость введения подобного правила замены ИЛИ-перехода объясняется тем, что любой переход, в том числе ИЛИ-переход, может сработать в рамках шага не более одного раза. Позиция-ограничитель обеспечивает выполнение этого условия. Из принципа одиночного срабатывания перехода внутри шага следует, что если на входе перехода имеется побудительный сигнал, а переход при этом не разрешен, то это означает, что переход уже мог сработать и побудительный сигнал можно удалить. На этом основано правило замены ИЛИ-перехода второго типа (рис. 3). Преимущество данного правила перед вышеприведенным состоит в том, что не требуется вычисление событийного графа. Это полезно при моделировании модульных NCES-сетей.

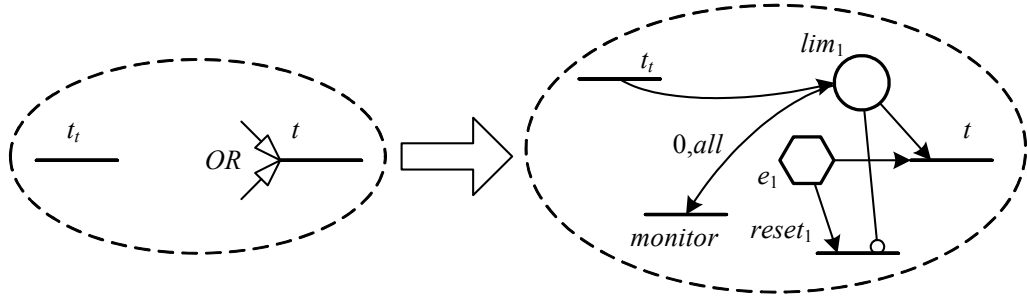


Рис. 2 Правило обработки ИЛИ-перехода (тип 1)

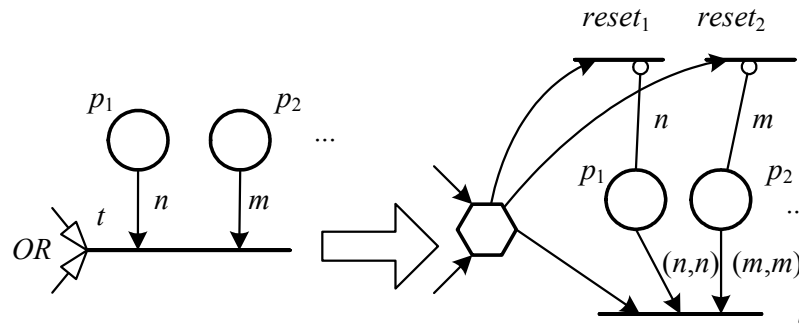


Рис. 3 Правило обработки ИЛИ-перехода (тип 2)

Каждому форсируемому И-переходу, имеющему более двух входящих событийных дуг, ставится в соответствие событийная позиция, дуга от этой событийной позиции к данному переходу с разметкой, равной (n,n) , где n – число входящих в переход событийных дуг, а также ингибиторная дуга от этой событийной позиции к переходу-монитору с разметкой, равной (n,all) . Схематично данное правило проиллюстрировано на рис. 4. Назовем событийные позиции, в которых может накапливаться более одной метки (подобных позиции e на рис. 4), порогово-событийными.

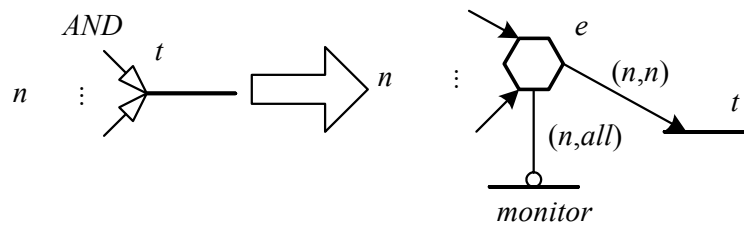


Рис. 4 Правило обработки И-перехода

Утверждение 2. Процесс вычисления и срабатывания разрешенного шага закончен тогда и только тогда, когда маркировка всех простых событийных позиций равна нулю, а маркировка порогово-событийных позиций меньше числа входящих в соответствующий И-переход событийных дуг.

Правило обработки перехода, увеличивающего маркировку позиции шага так, что становится разрешенным ранее не разрешенный переход шага, представлено на рис. 5. При этом каждой входной-выходной (внутренней)

позиции событийного графа ставится в соответствие новая позиция-двойник. Все дуги, ведущие во внутреннюю позицию из переходов событийного графа, перебрасываются в позицию-двойник. Для перемещения всех меток из позиции-двойника во внутреннюю позицию по окончании срабатывания шага создаются соответствующие дуги, связывающие позицию-двойник, переход-монитор и внутреннюю позицию.

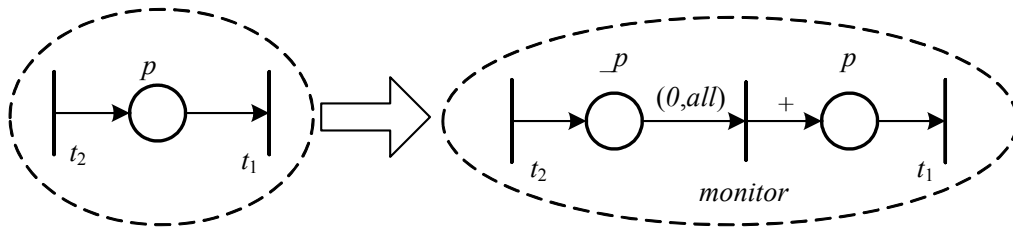


Рис. 5 Правило обработки перехода, из которого выходит дуга во внутреннюю позицию

Если в рамках одного событийного графа из разделяемой позиции выходит как обычная дуга, так и ингибиторная, то при срабатывании перехода, в который входит обычная дуга, может произойти такое изменение маркировки этой позиции, что переход, в который входит ингибиторная дуга, станет разрешенным и сможет при определенных условиях сработать, что противоречит определению шага. Если же из разделяемой позиции выходит не ингибиторная, а тестирующая дуга, то срабатывание перехода, в который входит обычная дуга, может привести к деактивации перехода, в который входит тестирующая дуга. Это также противоречит определению шага. Для решения данной проблемы предложено правило, приведенное на рис. 6. Использование данного правила расщепляет проблемную позицию на две. Маркировка основной позиции может изменяться при срабатывании переходов шага, в то время как маркировка позиции-двойника является неизменяемой на всем протяжении процесса срабатывания шага. Синхронизация маркировок этих двух позиций происходит при срабатывании триггерного перехода – источника сигнала.

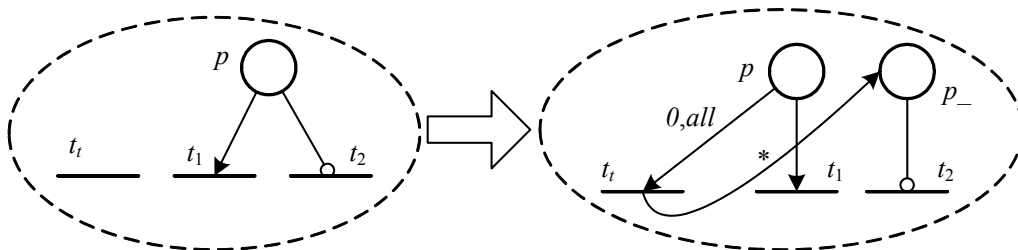


Рис. 6 Правило обработки разделяемой позиции, из которой выходят как обычные, так и ингибиторные (тестирующие) дуги

Модификация правила может заключаться в замене триггерного перехода t_i единым переходом-монитором *monitor*. Это бывает удобно в ряде случаев.

На рис. 7 представлено правило для разрешения ситуации, аккумулирующей ситуации, представленные в правилах 5 и 6. В данном случае, с од-

ной стороны, позиция p является внутренней, с другой – из данной позиции выходят как обычные, так и тестирующие дуги. Переход-монитор в данном правиле является единым для всей модели (режим 2).

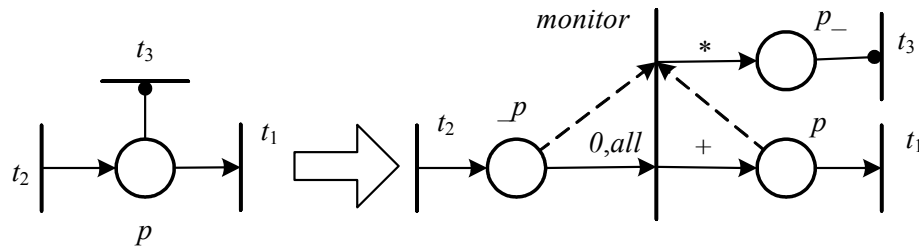


Рис. 7 Правило для разрешения комбинированной ситуации

Пример

В качестве примера рассмотрим преобразование модуля *NCES*-модели (рис. 8), представляющего вилку в задаче об обедающих философх [14]. Особенностью постановки данной задачи в работе [14] является ее трактовка в виде клиент-серверной системы. Асинхронная модель для указанного модуля приведена на рис. 9. Для преобразования использовались правила, представленные на рис. 6, 7, а также модификация правила на рис. 1, относящаяся к модульным сетям.

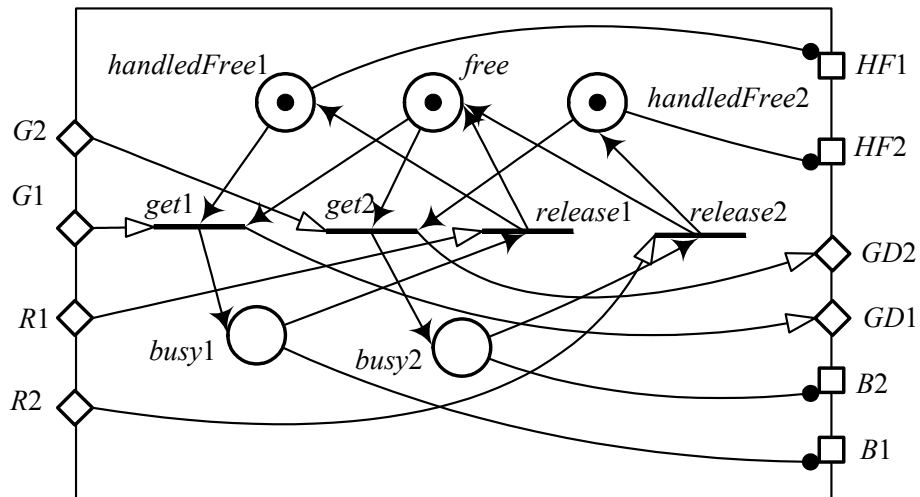


Рис. 8 *NCES*-модуль вилки

Заключение

В работе рассмотрены вопросы асинхронного моделирования *NCES*-сетей для их формальной верификации методом *Model Checking*. Направлением дальнейших исследований является развитие метода асинхронного моделирования для *aNCES*-сетей [4], а также разработка инженерной методики кодировки А-моделей на входном языке *SMV* и ее программная реализация в виде автоматического транслятора.

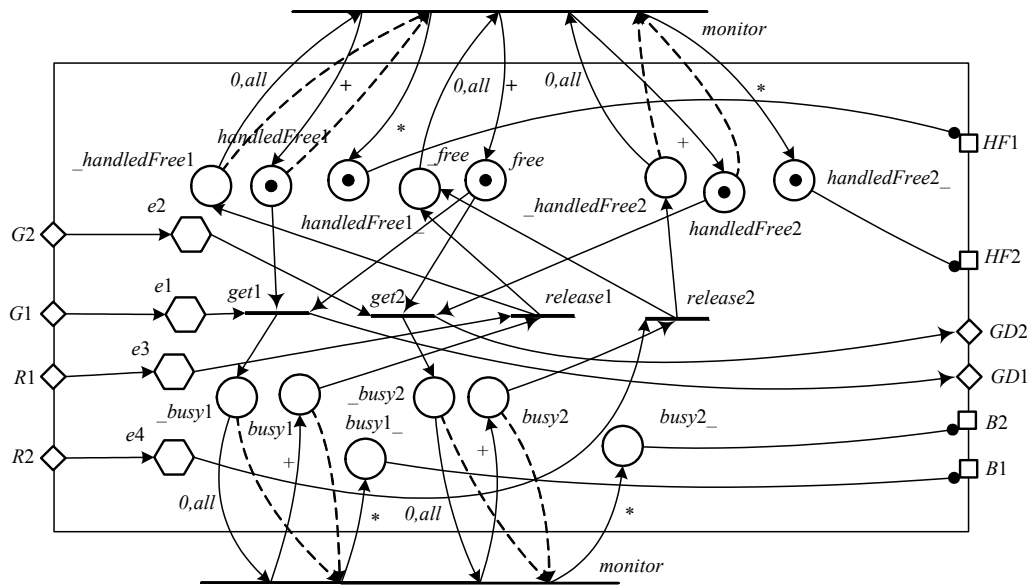


Рис. 9 А-модель NCES-модуля вилки

Список литературы

1. **Rausch, M.** Net condition/event systems with multiple condition outputs / M. Rausch, H.-M. Hanisch // IEEE Int. Conf. on Emerging Technologies and Factory Automation. – Paris, 1995. – Vol. 1. – P. 592–600.
2. **Starke, P. H.** Analysing Signal-Net Systems / P. H. Starke, S. Roch. Informatik-Bericht 162. – Berlin : Humboldt-Universitaet zu Berlin. – 2002. – 136 p.
3. ViVe – VisualVerifier Tool Framework <http://www.fb61499.com/license.html>
4. **Дубинин, В. Н.** Моделирование систем функциональных блоков ИЕС 61499 с помощью модульных арифметических NCES-сетей / В. Н. Дубинин // Новые информационные технологии и системы : труды VIII Международной научно-технической конференции. – Пенза, 2008. – Ч. 2. – С. 47–64.
5. **McMillan, K. L.** Symbolic Model Checking / K. L. McMillan. – Kluwer Academic Publishers, 1993.
6. **Кларк, Э.** Верификация моделей программ: Model Checking / Э. Кларк, О. Грамберг, Д. Пелед. – М. : МЦНМО, 2002. – 416 с.
7. **Вашкевич, Н. П.** Формализованное описание и верификация дискретных событийных систем с параллельными процессами / Н. П. Вашкевич, В. Н. Дубинин // Вопросы радиоэлектроники. – 2008. – Вып. 5 – С. 51–65. – (ЭВТ).
8. **Wimmel, G.** A BDD-based Model Checker for the PEP Tool / G. Wimmel // Major Individual Project Report, University of Newcastle. – Newcastle, 1997. – 107 p.
9. **Cortes, L. A.** Verification Methodology for Heterogeneous Hardware/Software Systems / L. A. Cortes, P. Eles, Z. Peng. – SAVE Project Report, Linköping University. – Linköping, 2000. – 18 p.
10. **Varea, M.** Symbolic Model Checking of Dual Transition Petri Nets / M. Varea, B. M. Al-Hashimi, L. A. Cortes, P. Eles, Z. Peng // Int. Symposium on Hardware/Software Codesign (CODES'02). – Colorado : Estes Park, 2002.
11. NuSMV – New Symbolic Model Checker <http://nusmv.iirst.itc.it>
12. **Dubin, V.** Event graph-based approach to interpretation of NCES models / V. Dubin, H.-M. Hanisch, D. Missal // Новые информационные технологии и системы : труды VII Международной научно-технической конференции. – Пенза, 2006. – Ч. 1. – С. 172–186.

13. **Ehrig, H.** Fundamentals of Algebraic Graph Transformation / H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. – Springer, 2006. – 373 с.
14. **Дубинин, В. Н.** Об одной методике проектирования супервизоров для предотвращения запрещенных состояний в дискретно-событийных системах / В. Н. Дубинин // Современные информационные технологии : труды Международной научно-технической конференции. – Пенза, 2008. – Вып. 8. – С. 143–159.

Дубинин Виктор Николаевич

кандидат технических наук, доцент,
кафедра вычислительной техники,
Пензенский государственный
университет

Dubinina Viktor Nikolaevich

Candidate of technical sciences,
associate professor, sub-department
of computer science, Penza State University

УДК 519.715

Дубинин, В. Н.

Асинхронное моделирование NCES-сетей / В. Н. Дубинин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 2 (10). – С. 3–14.