

Система автоматов: условия детерминизма и тестирование

И.Б. Бурдонов <igor@ispras.ru>

А.С. Косачев <kos@ispras.ru>

*Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. Статья посвящена проблеме тестирования составных систем, компоненты которых моделируются конечными автоматами с несколькими входами и выходами, а взаимодействие между ними – обменом сообщениями по симплексным каналам связи. Система описывается ориентированным графом связей, вершина которого соответствует автомату компонента, а дуга – каналу связи, соединяющему выход одного автомата со входом другого автомата. Предполагается выполненной следующая гипотеза о связях: граф связей статический, а отображаемая им структура связей не содержит ошибок. Автомат, находящийся в вершине графа, в каждом состоянии может принимать несколько сообщений по своим входам (не более одного по каждому входу) и посылать несколько сообщений по своим выходам (не более одного по каждому выходу). Определяется ассоциативная композиция автоматов системы. Показывается, при каких ограничениях на автоматы и граф связей их композиция, т.е. сама система, детерминирована. Целью тестирования является покрытие переходов автоматов компонентов, которые достижимы при работе этих автоматов в системе. Предполагается, что при тестировании возможно наблюдение изменения состояний автоматов системы и передаваемых между ними сообщений. Полное тестирование системы автоматов без учёта гипотезы о связях может потребовать число тестовых воздействий порядка произведения чисел состояний автоматов компонентов, а с учётом гипотезы о связях – порядка суммы этих чисел. При равном числе состояний всех автоматов это даёт экспоненциальное уменьшение числа тестовых воздействий. При условии выполнения гипотезы о связях для детерминированной системы предлагается алгоритм генерации тестов, основанный на фильтрации тестов, генерируемых для покрытия всех переходов композиции. Тест отбрасывается, если он покрывает только такие переходы в компонентах системы, которые покрываются остающимися тестами. В заключение определяются направления дальнейших исследований.

Ключевые слова: ориентированные графы, покрытие графа, взаимодействующие автоматы, композиция автоматов, детерминизм, распределенные системы, тестирование, сети.

DOI: 10.15514/ISPRAS-2016-28(1)-9

Для цитирования: Бурдонов И.Б., Косачев А.С. Система автоматов: условия детерминизма и тестирование. Труды ИСП РАН, том 28, вып. 1, 2016 г., с. 151-184. DOI: 10.15514/ISPRAS-2016-28(1)-9

1. Введение

Большинство сложных, особенно распределённых, систем представляет собой набор взаимодействующих компонентов. Данная статья продолжает исследование системы автоматов, начатое в нашей предыдущей статье [2]. Компоненты моделируются конечными автоматами, а взаимодействие – обменом сообщениями между автоматами. Автомат может иметь несколько входов для приёма сообщений и может принимать сразу несколько сообщений; также автомат может иметь несколько выходов для отправки сообщений и может посылать сразу несколько сообщений. Структура связей между компонентами моделируется ориентированным графом (будем называть его *графом связей*), вершинам которого соответствуют автоматы, а дуги называются *соединениями* и соответствуют симплексным каналам передачи сообщений. Соединение, ведущее из автомата *A* в автомат *B*, помечается выходом *j* автомата *A* и входом *i* автомата *B*. При этом каждый вход (выход) каждого автомата соединён не более чем с одним выходом (входом) другого (или того же самого) автомата. Входы (выходы) автоматов, которые не соединяются с выходами (входами) автоматов, являются *внешними*, через них осуществляется связь системы с её *окружением*. При тестировании тест подменяет собой окружение: он передаёт сообщения в систему на её внешние входы (набор таких сообщений можно назвать внешним *стимулом*) и принимает от системы сообщения с её внешних выходов (набор таких сообщений можно назвать внешней *реакцией*).

Считается, что граф связей статический, то есть не меняющийся в процессе работы системы. В этом случае система (также как её компоненты) может моделироваться конечным автоматом, получающимся из автоматов-компонентов с помощью подходящего оператора композиции, учитывающего граф связей.

Система работает правильно, если структура её связей правильна, и каждый автомат-компонент работает правильно. Обратное, вообще говоря, не верно, если требования к системе не однозначно определяют её структуру. Например, функциональные требования, определяющие внешнюю реакцию системы как требуемую функцию от последовательности внешних стимулов. В данной статье целью тестирования является покрытие всех переходов автоматов компонентов, которые достижимы при работе этих автоматов в системе. Поэтому, если в структуре связей нет ошибок, т.е. множество соединений совпадает с заданным (будем называть это *гипотезой о связях*), то такое тестирование системы сводится к проверке правильности переходов каждого автомата. Проблема в том, что каждый автомат может тестироваться только как часть системы. Это означает, что тест не имеет непосредственного

доступа к автомату-компоненту, и вынужден осуществлять тестовые воздействия с помощью сообщений, посылаемых по внешним входам, которые принадлежат, быть может, другим автоматам. Тестирование компонента такой системы похоже на тестирование в контексте ([3], [4], [5], [6]), когда этот компонент рассматривается как тестируемая система, а остальные – как контекст. Существенное отличие, однако, в том, что в таком контексте тоже могут быть ошибки, хотя, если верна гипотеза о связях, то только в компонентах, а не в структуре связей между ними. С другой стороны, один тест может проверять работу сразу нескольких компонентов, через которые проходят сообщения.

Поскольку компонент тестируется как часть системы, не все переходы автомата компонента, которые можно проверить при автономном тестировании с прямым доступом к компоненту, можно проверить при тестировании системы. Поэтому речь должна идти только о *достижимых переходах* автоматов, то есть таких переходах, которые могут быть выполнены при работе автомата как части системы.

Как мы показали в [1] на примере гипотеза о связях позволяет получить существенный выигрыш во времени тестирования, вплоть до экспоненциального его уменьшения. Если гипотеза о связях верна, то целью тестирования системы автоматов становится покрытие всех достижимых переходов автоматов компонентов системы.

В модели, предложенной в [2], автоматы системы работают синхронно: на каждом такте каждый автомат выполняет один переход. Также на каждом такте окружение системы посылает сообщения на некоторые внешние входы системы и принимает сообщения с некоторых внешних выходов системы. Для получения замкнутой формальной системы окружение моделируется автоматом, выходы которого соединяются с внешними входами системы, а внешние выходы системы соединяются со входами окружения. При тестировании роль окружения играет тест.

В данной статье предлагается алгоритм построения набора тестов, аналогичный алгоритму, предложенному в [1] для систем частного вида. Этот набор тестов является полным (проверяет все достижимые переходы автоматов компонентов) при выполнении двух условий: 1) верна гипотеза о связях, 2) система детерминирована. Дополнительно этот алгоритм определяет недостижимые переходы автоматов компонентов. Предполагается, во-первых, что нам известно, каким должен быть автомат каждого компонента (задан граф переходов автомата с точностью до изоморфизма) и именно это должно проверяться при тестировании. Во-вторых, тестовая система может не только выполнять тест, посылая сообщения на внешние входы системы и принимая сообщения с внешних выходов системы, но и наблюдать как состояния автоматов системы, так и сообщения, передаваемые по соединениям между автоматами системы. Такие предположения могут быть оправданы, например,

при имитационном тестировании аппаратуры (simulation-based verification) (см. например, [7]).

Сначала, в разделе 2, повторяются необходимые определения и утверждения (без доказательств) из [2]. Формально определяются модель автомата и модель системы, а также композиция переходов, композиция автоматов и композиция системы по соединению. Композиция автоматов удовлетворяет требованиям, предъявляемым к автомату, а композиция системы удовлетворяет требованиям, предъявляемым к системе автоматов. Композиция системы по всем её соединениям представляет собой автомат или набор автоматов, не связанных между собой соединениями. Композиция переходов, автоматов и системы обладает свойством ассоциативности. Ассоциативность важна для того, чтобы работа системы зависела только от множества её автоматов и соединений и не зависела от какого-либо упорядочивания этих множеств.

В разделе 3 даётся определение вполне определённого и детерминированного автомата, формулируются условия, при которых система таких автоматов тоже вполне определённая и детерминированная и доказывается соответствующая теорема.

В разделе 4 рассматриваются системы автоматов класса «вершины-дуги», которые являются обобщением системы автоматов, изучавшейся в [1].

Раздел 5 содержит описание алгоритма генерации полного набора тестов системы при выполнении гипотезы о связях.

В заключение намечаются направления дальнейших исследований.

2. Модель системы и композиция

В этом разделе мы повторим основные определения и утверждения из нашей предыдущей статьи [2].

Пусть задано некоторое конечное множество сообщений M , которое мы будем называть алфавитом (сообщений). Пустое множество $\emptyset$ будем называть пустым сообщением, считать, что $\emptyset \notin M$, и обозначать $M_\emptyset \triangleq M \cup \{\emptyset\}$.

Автомат в алфавите M – это набор $A = (M, I, J, S, T, s_0)$, где

I – конечное множество входов автомата,

J – конечное множество выходов автомата,

S – конечное множество состояний автомата,

$T \subseteq S \times X \times P \times Y \times Q \times S$ – множество переходов автомата, где

$X = \{x \mid x: I \rightarrow M_\emptyset\}$ – множество стимулов,

$Y = \{y \mid y: J \rightarrow M_\emptyset\}$ – множество реакций,

$P = 2^I$ – семейство подмножеств входов (по ним принимаются сообщения),

$Q = 2^J$ – семейство подмножеств выходов (по ним посылаются сообщения),

$s_0 \in S$ – начальное состояние,

причём выполнены следующие условия:

1. входы и выходы не пересекаются: $I \cap J = \emptyset$,
2. приниматься по входу и передаваться по выходу может только непустое сообщение: $\forall (s, x, p, y, q, t) \in T \ (\forall i \in p \ x(i) \neq \emptyset \ \& \ \forall j \in q \ y(j) \neq \emptyset)$,
что эквивалентно $\forall (s, x, p, y, q, t) \in T \ (p \subseteq x^{-1}(M) \ \& \ q \subseteq y^{-1}(M))$.

Очевидно, что из конечности множеств M , I , J и S следует конечность множеств X , Y , P , Q , T .

Для перехода $a = (s, x, p, y, q, t)$ состояние s будем называть *пресостоянием* перехода, а состояние t – *постсостоянием*, и будем обозначать $s_a = s$, $x_a = x$, $p_a = p$, $y_a = y$, $q_a = q$, $t_a = t$. Для автомата $A = (M, I, J, S, T, s_0)$ будем обозначать $I_A = I$, $J_A = J$, $S_A = S$, $T_A = T$, $s_{0A} = s_0$, а также $X_A = X$, $Y_A = Y$, $P_A = P$, $Q_A = Q$.

Пусть V – конечное множество автоматов в алфавите M . Без ограничения общности можно считать, что входы, выходы и состояния автоматов разные:

$\forall A, B \in V \ I_A \cap I_B = \emptyset \ \& \ I_A \cap J_B = \emptyset \ \& \ J_A \cap I_B = \emptyset \ \& \ J_A \cap J_B = \emptyset \ \& \ S_A \cap S_B = \emptyset$. Этого всегда можно добиться с помощью систематического переименования состояний, входов и выходов автоматов, в результате которого получаются автоматы изоморфные исходным.

Система автоматов – это набор $R = (M, V, E)$, где $E : E_{Dom} \rightarrow E_{Im}$ – биекция, определяющая *соединения*, где $E_{Dom} \subseteq I_R$, $E_{Im} \subseteq J_R$, $J_R = \bigcup \{J_A | A \in V\}$ – множество всех выходов всех автоматов, $I_R = \bigcup \{I_A | A \in V\}$ – множество всех входов всех автоматов. Для системы $R = (M, V, E)$ обозначим: $V_R = V$, $E_R = E$.

Определим функцию $\varphi : (I_R \cup J_R) \rightarrow V$, которая каждому входу или выходу системы R ставит в соответствие автомат, которому этот вход или выход принадлежит, т.е. $\forall A \in V \ \forall z \in I_A \cup J_A \ \varphi(z) = A$. Будем говорить, что соединение (j, i) ведёт из автомата $\varphi(j)$ в автомат $\varphi(i)$. Функция φ зависит от системы автоматов, однако для сокращения записи везде, где подразумевается система R , мы её не указываем в параметрах функции φ . Если функция φ применяется для другой системы $R' \neq R$, мы будем писать φ_R .

Вход $i \in I_R$, для которого не определено соединение, т.е. $i \notin E_{Im}$, будем называть *внешним входом системы*, и выход $j \in J_R$, для которого не определено соединение, т.е. $j \notin E_{Dom}$, будем называть *внешним выходом системы*.

Системе $R = (M, V, E)$ соответствует ориентированный граф с помеченными дугами, вершинами которого являются автоматы из V , а дуга $A \rightarrow B$ существует и помечена соединением (j, i) тогда и только тогда, когда $(j, i) \in E$, $\varphi(j) = A$ и $\varphi(i) = B$. Такой граф обозначим через $Og(R)$. Если снять ориентацию дуг, то соответствующий неориентированный граф обозначим через $Ng(R)$. Будем говорить, что автоматы A и B *связаны* в системе, если в графе $Ng(R)$ существует (неориентированный) путь из A в B (возможно, пустой, если $A = B$). В противном случае разные автоматы A и B *не связаны*. Очевидно, что отношение связности автоматов рефлексивно, симметрично и транзитивно, т.е. является отношением эквивалентности, которое разбивает множество автоматов V на классы эквивалентности, которые вместе с соединениями

между автоматами одного класса мы будем называть *компонентами связности*.

Соединение $(j,i) \in E$ будем называть *петлёй*, если это дуга-петля в графе $\mathbf{Og}(\mathbf{R})$, то есть $\varphi(j) = \varphi(i)$.

Для удобства будем считать, что каждое состояние каждого автомата в системе $\mathbf{R} = (M, V, E)$ является множеством, причём состояния разных автоматов в системе являются непересекающимися множествами: $\forall A, B \in V \quad \forall s_A \in S_A \quad \forall s_B \in S_B \quad (A \neq B \Rightarrow s_A \cap s_B = \emptyset)$. Если уже задана система, в которой это условие не выполнено, то можно преобразовать каждый автомат, заменив в нём каждое состояние s на синглетон $\{s\}$. Поскольку в системе автоматов состояния разных автоматов разные, то таким преобразованием будут получены автоматы изоморфные исходным, а для системы автоматов будет выполнено условие попарного непересечения множеств состояний автоматов.

Введём способ сокращённой записи для преобразования функции с помощью уменьшения её домена: для произвольной функции f и произвольного множества N обозначим $f/N \triangleq f \setminus \{(z, f(z)) \mid z \in N \cap \mathbf{Dom}(f)\}$. Очевидно, что для произвольного множества N имеет место $f/(N \cap N') = (f/N)/N'$ и для произвольной функции g такой, что $\mathbf{Dom}(f) \cap \mathbf{Dom}(g) = \emptyset$ и $N \cap \mathbf{Dom}(g) = \emptyset$, имеет место $(f \cup g)/N = (f/N) \cup g$. Будем считать, что операция “/” имеет тот же приоритет, что разность “\” и объединение “ $\cup$ ” множеств. Как следствие, в выражениях над множествами, где используются только операции “/”, “\” и “ $\cup$ ”, мы будем использовать бесскобочную запись и предполагать, что операции выполняются слева направо.

Логическую эквивалентность будем обозначать символом « $\sim$ »: $a \sim b \triangleq (a \& b) \vee (\neg a \& \neg b)$.

Композиция определяется по соединению $(j,i) \in E$. Обозначим через A автомат, которому принадлежит выход j , т.е. $A = \varphi(j)$, а через B автомат, которому принадлежит вход i , т.е. $B = \varphi(i)$.

Композиция переходов по соединению. Для соединения (j,i) определим условие $f(a,j,i,b)$ композиции двух переходов a и b и результат композиции $a[j,i]b$:

$$f(a,j,i,b) \triangleq a \in T_A \& b \in T_B \& (A=B \Rightarrow a=b) \& y_a(j) = x_b(i) \& (j \in q_a \sim i \in p_b),$$

$$f(a,j,i,b): a[j,i]b \triangleq (s_a \cup s_b, x_a \cup x_b \setminus \{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b \setminus \{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b).$$

Заметим, что если $A=B$, то

$$f(a,j,i,a) \triangleq a \in T_A \& y_a(j) = x_a(i) \& (j \in q_a \sim i \in p_a),$$

$$f(a,j,i,a): a[j,i]a \triangleq (s_a, x_a \setminus \{i\}, p_a \setminus \{i\}, y_a \setminus \{j\}, q_a \setminus \{j\}, t_a).$$

Композиция переходов естественным образом распространяется на композицию множеств переходов, определяемую как множество попарных

композиций переходов. Для соединения (j,i) определим композицию множеств переходов G и H : $G[j,i]H \triangleq \{a[j,i]b | a \in G \ \& \ b \in H \ \& \ f(a,j,i,b)\}$.

Композиция автоматов по соединению.

Для множества переходов T обозначим множество пре- и постсостояний этих переходов: $States(T) \triangleq \{s_a | a \in T\} \cup \{t_a | a \in T\}$. Для соединения (j,i) определим композицию автоматов A и B :

$$A[j,i]B \triangleq (M, I_A \cup I_B \setminus \{i\}, J_A \cup J_B \setminus \{j\}, \{s_{0A} \cup s_{0B}\} \cup States(T_A[j,i]T_B), T_A[j,i]T_B, s_{0A} \cup s_{0B}).$$

Композиция системы по соединению. Для системы R и соединения (j,i) определим $R[j,i] \triangleq (M, V \setminus \{A,B\} \cup \{A[j,i]B\}, E \setminus \{(j,i)\})$.

В [2] доказано, что композиция автоматов и системы определены корректно, т.е. композиция автоматов удовлетворяет условиям, налагаемым на автомат (теорема 1 в [2]), а композиция системы удовлетворяет условиям, налагаемым на систему (теорема 2 в [2]).

Последовательностью соединений для системы $R = (M, V, E)$ назовём последовательность Z соединений из E , в которой каждое соединение встречается не более одного раза, т.е. $Z: [1..|Z|] \rightarrow E$ является инъекцией. Определим композицию системы R по последовательности Z как $R[Z] \triangleq R[Z(1)][Z(2)]...[Z(|Z|)]$, если Z не пусто, и $R[Z] \triangleq R$, если Z пусто.

Последовательность соединений, являющуюся биекцией, т.е. содержащую каждое соединение из E , будем называть *последовательностью всех соединений* и обозначать $Z^\wedge$. Очевидно, что в системе $R[Z^\wedge]$ нет соединений, т.е. $E_{R[Z^\wedge]} = \emptyset$, т.е. все входы и выходы автоматов являются внешними. Если в системе R были автоматы, которые не связаны друг с другом, то в $R[Z^\wedge]$ окажется несколько автоматов, соответствующих компонентам связности в R , и все эти автоматы также не связаны друг с другом.

В [2] доказана ассоциативность композиции переходов (лемма 1 в [2]), автоматов (лемма 2 в [2]) и системы (лемма 3 в [2]), т.е. независимость результата композиции по двум соединениям $(j,i) \in E$ и $(l,k) \in E$ от порядка композиции: сначала по (j,i) , а потом по (l,k) , или, наоборот, сначала по (l,k) , а потом по (j,i) . Как следствие, композиция системы R по всем её соединениям не зависит от порядка композиции (теорема 3 в [2]): для любых двух последовательностей всех соединений $Z^\wedge$ и $Z'^\wedge$ имеет место $R[Z^\wedge] = R[Z'^\wedge]$. Ассоциативность важна для того, чтобы работа системы зависела только от множества её автоматов и соединений и не зависела от какого-либо упорядочивания этих множеств.

3. Вполне определённости и детерминизм

В композиции, которая определена выше, нет «асинхронных» переходов, все переходы «синхронные»: переход в каждом автомате выполняется одновременно с переходами в автоматах, связанных с ним по соединениям.

Это означает, что если какой-то автомат «стоит», т.е. в текущем своем состоянии не может выполнить никакого перехода, то «стоят» все автоматы из того же компонента связности, и это «стояние» продолжается бесконечно до рестарта системы. Это ситуация тупика (deadlock).

Тупик возникает из-за несогласованности требований автоматов, связанных соединением. Эти требования определяются условиями композиции переходов $a[j,i]b$: 1) $y_a(j)=x_b(i)$ и 2) $j \in q_a \sim i \in p_b$. Для того чтобы тупик был невозможен из-за нарушения условия 1, достаточно потребовать, чтобы либо А) в каждом автомате в каждом состоянии был определен переход по каждому стимулу x , либо В) в каждом автомате в каждом состоянии был определен переход по каждой реакции y . Для того чтобы тупик был невозможен из-за нарушения условия 2, достаточно потребовать, чтобы либо А) в каждом автомате в каждом состоянии для каждой реакции y был определен переход с каждым возможным параметром q , определяющим по какому выходу j сообщение $y(j)$ передаётся, а по какому нет, либо В) в каждом автомате в каждом состоянии для каждого стимула x был определен переход с каждым параметром p , определяющим по какому входу i сообщение $x(i)$ принимается, а по какому нет. Мы выбираем варианты А для обоих условий, что формализовано в определении вполне определенного автомата.

Автомат $A=(M,I,J,S,T,s_0)$ будем называть *определённым по всем стимулам*, если в каждом состоянии есть переход по каждому стимулу:

$$1. \quad \forall s \in S \quad \forall x \in X \quad \exists a \in T \quad s_a = s \quad \& \quad x_a = x,$$

Автомат $A=(M,I,J,S,T,s_0)$ будем называть *вполне определенным*, если он определен по всем стимулам и есть переходы по всем возможным q :

$$2. \quad \forall a \in T \quad \forall q \in \subseteq_y^{-1}(M) \quad \exists a' \in T \quad s_{a'} = s_a \quad \& \quad x_{a'} = x_a \quad \& \quad p_{a'} = p_a \quad \& \quad y_{a'} = y_a \quad \& \quad q_{a'} = q'.$$

Вместо условий 1 и 2 можно использовать *правила умолчания*, если в определении автомата указывать подмножество T^* явных переходов, и дополнять его до множества T всех переходов согласно правилам умолчания:

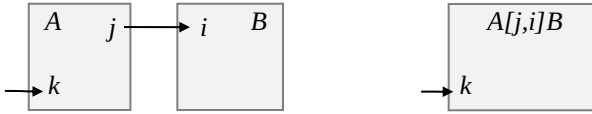
- 1) Если T^* не содержит явного перехода в некотором состоянии по некоторому стимулу, то это означает, что такой переход определён неявно без приёма и выдачи сообщений и без изменения состояния.
- 2) Если в T^* есть переход (s, x, p, y, q, t) , но для некоторого $q' \in \subseteq_y^{-1}(M)$ нет перехода вида (s, x, p, y, q', t') , то это означает, что такой переход определён неявно без изменения состояния ($t'=t$).

Формально это означает:

$$T \triangleq T^* \cup \{(s, x, \emptyset, J \times \{ \Lambda \}, \emptyset, s) \mid s \in S \quad \& \quad x \in X \quad \& \quad \forall p, y, q, t \quad (s, x, p, y, q, t) \notin T^*\} \\ \cup \{(s, x, p, y, q', s) \mid \exists q, t \quad (s, x, p, y, q, t) \in T^* \quad \& \quad q' \in \subseteq_y^{-1}(M) \quad \& \quad \forall t' \quad (s, x, p, y, q', t') \notin T^*\}.$$

В некотором смысле неявный переход вида $(s, x, \emptyset, J \times \{ \Lambda \}, \emptyset, s)$ моделирует отсутствие перехода, поскольку состояние автомата не меняется, сообщения не принимаются и не передаются.

Заметим, что, если в определении автомата используются правила умолчания, то композиция автоматов должна определяться через композицию не только явных, но и неявных переходов. Это показано в примере на Рис. 1.



$$A = (\{m\}, \{k\}, \{j\}, \{s_{A0}\}, T_A^*, \{s_{A0}, s_{A1}\})$$

$$T_A = T_A^* = \{ (s_{A0}, \{(k, A)\}, \emptyset, \{(j, A)\}, \emptyset, s_{A0}), (s_{A0}, \{(k, m)\}, \{k\}, \{(j, A)\}, \emptyset, s_{A1}) \}$$

$$B = (\{m\}, \{i\}, \emptyset, \{s_{B0}\}, T_B^*, s_{B0})$$

$$T_B^* = \emptyset, T_B = \{ (s_{B0}, \{(i, A)\}, \emptyset, \emptyset, \emptyset, s_{B0}), (s_{B0}, \{(i, m)\}, \emptyset, \emptyset, \emptyset, s_{B0}) \}$$

$$T_{A[j,i]}T_B = \{ (s_{A0} \cup s_{B0}, \{(k, A)\}, \emptyset, \emptyset, \emptyset, s_{A0} \cup s_{B0}), (s_{A0} \cup s_{B0}, \{(k, m)\}, \{k\}, \emptyset, \emptyset, s_{A1} \cup s_{B0}) \} = T_{A[j,i]B}$$

$T_{A[j,i]}T_B^* = \emptyset$, а правила умолчания добавляют множество переходов:

$$\{ (s_{A0} \cup s_{B0}, \{(k, A)\}, \emptyset, \emptyset, \emptyset, s_{A0} \cup s_{B0}), (s_{A0} \cup s_{B0}, \{(k, m)\}, \emptyset, \emptyset, \emptyset, s_{A0} \cup s_{B0}) \} \neq T_{A[j,i]B}$$

Рис. 1. Композиция и неявные переходы

Fig 1. Composition and implicit transitions

В этом примере автомат B не имеет явных переходов, поэтому все его переходы неявные и являются петлями в начальном состоянии s_{B0} . Из-за этого композиция $T_{A[j,i]}T_B^*$ множеств явных переходов пуста, а после её пополнения по правилам умолчания все добавляемые переходы будут петлями в начальном состоянии $s_{A0} \cup s_{B0}$. В то же время композиция $T_{A[j,i]}T_B$ множеств всех переходов (как явных, так и неявных) не требует пополнения по правилам умолчания, но содержит переход по стимулу $x = \{(k, m)\}$, который ведёт в состояние $s_{A1} \cup s_{B0}$ отличное от начального.

Для целей тестирования мы ограничимся исследованием детерминированных автоматов. Детерминированность будет означать, что каждый автомат в системе в своём текущем состоянии может выполнить не более одного перехода. При условии вполне определенности будет ровно один переход. Для детерминизма, прежде всего, нужно, чтобы в каждом состоянии автомата каждый стимул x однозначно определял параметр p , то есть по какому входу i сообщение $x(i)$ принимается, а по какому нет. Стимул определяется сообщениями на входах автомата, а они, в свою очередь, определяются автоматами, посылающими сообщения на входы данного автомата, то есть автоматами, выходы которых соединены со входами данного автомата. Кроме того, стимул x однозначно определяет реакцию y . При этом параметр q ,

определяющий по какому выходу j сообщение $y(j)$ передаётся, а по какому нет, может выбираться любой, но зато однозначно определяется автоматами, принимающими сообщения с выходов данного автомата, то есть автоматами, входы которых соединены с выходами данного автомата. Наконец, постсостояние t должно быть единственным, если определены остальные параметры перехода: s , x , p , u и q . Эти требования формализованы ниже в определении детерминированного автомата.

Автомат $A=(M,I,J,S,T,s_0)$ будем называть *детерминированным*, если выполнены условия:

1. Состояние s и стимул x однозначно определяют приём сообщений p и реакцию u :

$$\forall a,b \in T_A (s_a=s_b \ \& \ x_a=x_b \Rightarrow p_a=p_b \ \& \ u_a=u_b).$$

2. Одинаково помеченные переходы, ведущие из одного состояния, совпадают, т.е. ведут в одно постсостояние:

$$\forall a,b \in T_A (s_a=s_b \ \& \ x_a=x_b \ \& \ p_a=p_b \ \& \ u_a=u_b \ \& \ q_a=q_b \Rightarrow t_a=t_b).$$

Заметим, что если выполнено условие 1, условие 2 можно переписать короче:

$$\forall a,b \in T_A (s_a=s_b \ \& \ x_a=x_b \ \& \ q_a=q_b \Rightarrow t_a=t_b).$$

В то же время детерминированность каждого автомата системы не гарантирует детерминированность системы: без дополнительных условий автоматы в некотором цикле соединений могут выполнять либо один, либо другой набор переходов, выбор между которыми недетерминирован. Выяснению достаточных условий детерминизма системы как раз и посвящена основная часть данной статьи.

Пусть задана система автоматов $R=(M,V,E)$.

Лемма 1 о композиции разных автоматов:

Композиция различных вполне определённых и детерминированных автоматов вполне определённая и детерминированная.

Доказательство: Пусть автоматы $A,B \in V$ вполне определённые, детерминированные и различные $A \neq B$, а соединение $(j,i) \in E$ ведёт из A в B . Надо доказать, что автомат $A[j,i]B$ вполне определённый и детерминированный.

1. Докажем, что автомат $A[j,i]B$ вполне определённый.

- 1.1. Докажем выполнение 1-го условия вполне определённости (определённость по всем стимулам):

$$\forall s \in S_{A[j,i]B} \ \forall x \in X_{A[j,i]B} \ \exists c \in T_{A[j,i]B} \ s_c=s \ \& \ x_c=x.$$

Пусть $s \in S_{A[j,i]B}$ и $x \in X_{A[j,i]B}$. Тогда найдутся такие состояния $s_A \in S_A$ и $s_B \in S_B$, что $s=s_A \cup s_B$. Поскольку $\text{Dom}(x)=I_{A[j,i]B}=I_A \cup I_B \setminus \{i\}$ и $i \in I_B$, стимул x может быть представлен в виде объединения непересекающихся по доменам отображений $x=x_A \cup x_B$, где $\text{Dom}(x_A)=I_A$ и $\text{Dom}(x_B)=I_B \setminus \{i\}$. Поскольку автомат A вполне определённый, по условию 1 в нём есть переход по каждому стимулу в каждом состоянии. Следовательно, имеется переход a

в состоянии s_A по стимулу x_A , т.е. $s_a = s_A$ и $x_a = x_A$. Объединение $x'_B = x_B \cup \{(i, y_a(j))\}$ является стимулом вполне определённого автомата B , поэтому по условию 1 имеется переход b в состоянии s_B по стимулу x'_B , т.е. $s_b = s_B$ и $x_b = x'_B$. Поскольку автомат A вполне определённый, по условию 2 в нём имеются переходы по всем q , следовательно, найдётся в состоянии s_a такой переход a' , что $s_{a'} = s_a$, $x_{a'} = x_a$ и $j \in q_a \sim i \in p_b$. Тогда $f(a', j, i, b) = \text{true}$. Следовательно, в автомате $A[j, i]B$ имеется в состоянии $s_a \cup s_b$ переход $c = a'[j, i]b$ по стимулу $x_a \cup x_b \setminus \{i\}$. Поскольку $s = s_A \cup s_B$, $s_a = s_A$, $s_{a'} = s_a$, $s_b = s_B$, имеем $s_c = s_a \cup s_b = s$. Поскольку $x = x_A \cup x_B$, $x'_B = x_B \cup \{(i, y_a(j))\}$, $x_b = x'_B$, $x_a = x_A$, $x_{a'} = x_a$, имеем $x_c = x_a \cup x_b \setminus \{i\} = x$. Тем самым, в автомате $A[j, i]B$ имеется в состоянии s переход c по стимулу x , что и требовалось доказать.

1.2. Докажем выполнение 2-го условия вполне определённости:

$$\forall c \in T_{A[j, i]B} \quad \forall q' \subseteq_{\mathcal{U}_c}^{-1}(M) \quad \exists (s_c, x_c, p_c, y_c, q', t') \in T_{A[j, i]B}.$$

Пусть $c \in T_{A[j, i]B}$ и $q' \subseteq_{\mathcal{U}_c}^{-1}(M)$. По определению композиции переход c является композицией двух переходов $c = a[j, i]b$, где $a \in T_A$ и $b \in T_B$. Тогда $y_c = y_a \cup y_b \setminus \{j\}$. Поскольку $\text{Dom}(y_a) = J_A$ и $\text{Dom}(y_b) = J_B$, а $q' \subseteq_{\mathcal{U}_c}^{-1}(M)$, множество q' можно представить в виде объединения двух множеств $q' = q_A \cup q_B$, где $q_A \subseteq_{\mathcal{U}_a}^{-1}(M) \setminus \{j\}$ и $q_B \subseteq_{\mathcal{U}_b}^{-1}(M)$. Обозначим $q'_A = q_A$, если $i \notin p_b$, и $q'_A = q_A \cup \{j\}$, если $i \in p_b$. Поскольку автоматы A и B вполне определённые по условию 2 найдутся такие состояния $t_A \in S_A$ и $t_B \in S_B$, что $a' = (s_a, x_a, p_a, y_a, q'_A, t_A) \in T_A$ и $b' = (s_b, x_b, p_b, y_b, q_B, t_B) \in T_B$. Имеем $y_a(j) = y_a(i) = x_b(i) = x_b(j)$ и $j \in q_a \sim j \in q'_A \sim i \in p_b \sim i \in p_b$. Следовательно, $f(a', j, i, b) = \text{true}$. Также имеем $q' = q_A \cup q_B = q'_A \cup q_B \setminus \{j\}$. Поэтому, обозначая $c' = a'[j, i]b'$, имеем $c' = (s_c, x_c, p_c, y_c, q', t_A \cup t_B) \in T_{A[j, i]B}$, т.е. в композиции нашлось такое состояние $t' = t_A \cup t_B$, что $(s_c, x_c, p_c, y_c, q', t') \in T_{A[j, i]B}$, что и требовалось доказать.

2. Докажем, что автомат $A[j, i]B$ детерминированный.

По определению композиции автоматов переход в автомате $A[j, i]B$ получается в результате композиции двух переходов в автоматах A и B . Пусть $a, a' \in T_A$, $b, b' \in T_B$ и переход a компонуется с переходом b , а переход a' – с переходом b' , т.е. $f(a, j, i, b) = f(a', j, i, b) = \text{true}$. Докажем выполнение условий детерминизма для переходов $a[j, i]b$ и $a'[j, i]b'$. По определению композиции переходов для $a, a' \in T_A$, $b, b' \in T_B$ и $A \neq B$ имеем:

$$f(a, j, i, b) \Rightarrow y_a(j) = x_b(i) \text{ \& } (j \in q_a \sim i \in p_b),$$

$$f(a', j, i, b') \Rightarrow y_{a'}(j) = x_{b'}(i) \text{ \& } (j \in q_{a'} \sim i \in p_{b'}),$$

$$a[j, i]b = (s_a \cup s_b, x_a \cup x_b \setminus \{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b \setminus \{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b) \text{ и}$$

$$a'[j, i]b' = (s_{a'} \cup s_{b'}, x_{a'} \cup x_{b'} \setminus \{i\}, p_{a'} \cup p_{b'} \setminus \{i\}, y_{a'} \cup y_{b'} \setminus \{j\}, q_{a'} \cup q_{b'} \setminus \{j\}, t_{a'} \cup t_{b'}).$$

$$\text{Пусть } s_a \cup s_b = s_{a'} \cup s_{b'} \text{ и } x_a \cup x_b \setminus \{i\} = x_{a'} \cup x_{b'} \setminus \{i\}.$$

2.1. Докажем выполнение 1-го условия детерминизма:

$$p_a \cup p_b \setminus \{i\} = p_a \cup p_b \setminus \{i\} \text{ и } y_a \cup y_b \setminus \{j\} = y_a \cup y_b \setminus \{j\}.$$

$A \neq B$ влечёт $s_a \cap s_b = s_a \cap s_b = \emptyset$. Поэтому, $s_a \cup s_b = s_a \cup s_b$, влечёт $s_a = s_a$ и $s_b = s_b$. $A \neq B$ влечёт $I_A \cap I_B = \emptyset$, что влечёт $x_a \cap x_b = x_a \cap x_b = \emptyset$. Также $i \in I_B$. Поэтому $x_a \cap x_b \setminus \{i\} = x_a \cap x_b \setminus \{i\}$ влечёт $x_a = x_a$ и $x_b \setminus \{i\} = x_b \setminus \{i\}$. По 1-ому условию детерминизма для A из $s_a = s_a$ и $x_a = x_a$ следует $y_a = y_a$ и $p_a = p_a$. Поскольку $y_a(j) = x_b(i)$ и $y_a(j) = x_b(i)$, а $y_a = y_a$ влечёт $y_a(j) = y_a(j)$, имеем $x_b(i) = x_b(i)$. А тогда, поскольку $x_b \setminus \{i\} = x_b \setminus \{i\}$, имеем $x_b = x_b$. По 1-ому условию детерминизма для B из $s_b = s_b$ и $x_b = x_b$ следует $y_b = y_b$ и $p_b = p_b$. Итак, $s_a = s_a$, $s_b = s_b$, $x_a = x_a$, $x_b = x_b$, $p_a = p_a$, $p_b = p_b$, $y_a = y_a$, $y_b = y_b$. А тогда $p_a \cup p_b \setminus \{i\} = p_a \cup p_b \setminus \{i\}$ и $y_a \cup y_b \setminus \{j\} = y_a \cup y_b \setminus \{j\}$, что и требовалось доказать.

2.2. Пусть также $q_a \cup q_b \setminus \{j\} = q_a \cup q_b \setminus \{j\}$. Докажем выполнение 2-го условия детерминизма: $t_a \cup t_b = t_a \cup t_b$.

По доказанному $s_a = s_a$, $s_b = s_b$, $x_a = x_a$, $x_b = x_b$, $p_a = p_a$, $p_b = p_b$, $y_a = y_a$, $y_b = y_b$. $A \neq B$ влечёт $J_A \cap J_B = \emptyset$, что влечёт $q_a \cap q_b = q_a \cap q_b = \emptyset$. Также $j \in J_A$. Поэтому $q_a \cup q_b \setminus \{j\} = q_a \cup q_b \setminus \{j\}$ влечёт $q_a \setminus \{j\} = q_a \setminus \{j\}$ и $q_b = q_b$. Поскольку $j \in q_a \sim i \in p_b$ и $j \in q_a \sim i \in p_b$, а $p_b = p_b$, имеем $j \in q_a \sim j \in q_a$. А поскольку $q_a \setminus \{j\} = q_a \setminus \{j\}$, имеем $q_a = q_a$. Итак, $s_a = s_a$, $s_b = s_b$, $x_a = x_a$, $x_b = x_b$, $p_a = p_a$, $p_b = p_b$, $y_a = y_a$, $y_b = y_b$, $q_a = q_a$, $q_b = q_b$. А тогда по 2-ому условию детерминизма для A и для B имеем $t_a = t_a$ и $t_b = t_b$. А тогда $t_a \cup t_b = t_a \cup t_b$, что и требовалось доказать.

Лемма 1 доказана.

Пару (k, l) будем называть *контактом* внутри автомата A , если k вход, а l выход автомата A : $l \in J_A$ & $k \in I_A$. Контакт (k, l) будем называть *свободным* и обозначать $k \# l$, если сообщение на выходе l не зависит от сообщения на входе k :

$$k \# l \triangleq \exists A \in V \ l \in J_A \ \& \ k \in I_A \ \& \ \forall a, b \in T_A (s_a = s_b \ \& \ x_a \setminus \{k\} = x_b \setminus \{k\} \Rightarrow y_a(l) = y_b(l)).$$

Очевидно, что свобода контакта внутри автомата является свойством самого этого автомата независимо от того, в какую систему он входит. Однако принадлежность выхода и входа тому или иному автомату зависит от системы и меняется при композиции системы, поскольку меняется множество автоматов системы. Если $l \in J_A$, $k \in I_A$ и выполняется композиция автоматов $A[j, i]B$ или $B[j, i]A$, то выход l остаётся в системе, только если $l \neq j$, вход k остаётся в системе, только если $k \neq i$. Кроме того, если $l \neq j$ и $k \neq i$, то после композиции выход l и вход k будут принадлежать уже другому, композиционному автомату $A[j, i]B$ или $B[j, i]A$. Поэтому, если речь идёт о системе R отличной от подразумеваемой системы R , мы в обозначении будем указывать систему в индексе: $k \#_R l$.

Лемма 2 о композиции по петле:

Если автомат вполне определённый и детерминированный, внутри автомата контакт $i \# j$ и в системе есть соединение-петля (j, i) , то композиция этого автомата по этому соединению вполне определённая и детерминированная.

Доказательство: Пусть соединение $(j,i) \in E$ является петлёй, т.е. $\varphi(j) = \varphi(i)$, контакт $i \neq j$, и автомат $A = \varphi(j)$ вполне определённый и детерминированный.

1. Докажем, что автомат $A[j,i]A$ вполне определённый.

1.1. Докажем выполнение 1-го условия вполне определённости (определённость по всем стимулам):

$$\forall s \in S_{A[j,i]A} \quad \forall x \in X_{A[j,i]A} \quad \exists c \in T_{A[j,i]A} \quad s_c = s \text{ \& } x_c = x.$$

Пусть $s \in S_{A[j,i]A}$ и $x \in X_{A[j,i]A}$. Тогда $s \in S_A$. Поскольку $Dom(x) = I_{A[j,i]A} = I_A \setminus \{i\}$, для любого сообщения $m \in M_A$ объединение $x' = x \cup \{(i, m)\}$ является стимулом для автомата A . Поскольку автомат A вполне определённый, по условию 1 в нём есть переход по каждому стимулу в каждом состоянии. Следовательно, имеется переход a в состоянии s по стимулу x' , т.е. $s_a = s$ и $x_a = x'$. Тогда $x'' = x \cup \{(i, y_a(j))\}$ тоже является стимулом для автомата A . Поскольку автомат A вполне определённый, по условию 1 в нём есть переход a' в состоянии s по стимулу x'' , т.е. $s_{a'} = s$ и $x_{a'} = x''$. Очевидно, что $x_a(i) = y_a(j)$ и $x_{a'} \setminus \{i\} = x_a \setminus \{i\}$. А тогда, поскольку $i \neq j$, имеем $x_a(i) = y_a(j) = y_{a'}(j)$, что влечёт $x_{a'}(i) = y_{a'}(j)$. Поскольку автомат A вполне определённый, по условию 2 в нём имеются переходы по всем q , следовательно, найдётся в состоянии s переход $a'' = (s_a, x_a, p_a, y_a, q_a, t_a)$, где $q_a \setminus \{i\} = q_{a'} \setminus \{i\}$ и $j \in q_a \setminus \sim i \in p_{a''}$. Тогда $f(a'' \setminus j, i, a'') = \text{true}$. Следовательно, в автомате $A[j,i]A$ имеется в состоянии s переход $c = a''[j,i]a''$ по стимулу $x_a \setminus \{i\}$. Поскольку $s_{a''} = s_a = s$, имеем $s_c = s$. Поскольку $x_{a''} = x_a = x' = x \cup \{(i, y_a(j))\}$, имеем $x_c = x_a \setminus \{i\} = x$. Тем самым, в автомате $A[j,i]A$ имеется в состоянии s переход c по стимулу x , что и требовалось доказать.

1.2. Докажем выполнение 2-го условия вполне определённости:

$$\forall c \in T_{A[j,i]A} \quad \forall q' \sqsubseteq_{y_c}^{-1}(M) \quad \exists (s_c, x_c, p_c, y_c, q', t') \in T_{A[j,i]A}.$$

Пусть $c \in T_{A[j,i]A}$ и $q' \sqsubseteq_{y_c}^{-1}(M)$. По определению композиции переход c является композицией одного перехода $c = a[j,i]a$, где $a \in T_A$. Тогда $x_a(i) = y_a(j)$, $j \in q_a \sim i \in p_a$ и $y_c = y_a \setminus \{j\}$, $s_c = s_a$. Обозначим $q'' = q'$, если $i \notin p_a$, и $q'' = q' \cup \{j\}$, если $i \in p_a$. Поскольку автомат A вполне определённый, по условию 2 найдётся переход $a' = (s_c, x_a, p_a, y_a, q'', t_a) \in T_A$. Имеем $y_a(j) = y_a(j) = x_a(i) = x_{a'}(i)$ и $j \in q_a \setminus \sim j \in q'' \sim i \in p_a \sim i \in p_{a'}$. Следовательно, $f(a' \setminus j, i, a') = \text{true}$. Также имеем $q' = q'' \setminus \{j\} = q_a \setminus \{j\}$. Поэтому, обозначая $c' = a'[j,i]a'$, имеем $c' = (s_c, x_c, p_c, y_c, q', t_a) \in T_{A[j,i]A}$, т.е. в композиции нашлось такое состояние $t' = t_a$, что $(s_c, x_c, p_c, y_c, q', t') \in T_{A[j,i]A}$, что и требовалось доказать.

2. Докажем, что автомат $A[j,i]A$ детерминированный.

По определению композиции автоматов переход в автомате $A[j,i]A$ получается в результате композиции одного перехода в автомате A . Пусть $a, b \in T_A$ и каждый из переходов a и b компонуется сам с собой, т.е. $f(a, j, i, a) = f(b, j, i, b) = \text{true}$. Надо доказать выполнение обоих условий

детерминизма для переходов $a[j,i]a$ и $b[j,i]b$. По определению композиции переходов для $a, b \in T_A$ имеем:

$$f(a, j, i, a) \Rightarrow y_a(j) = x_a(i) \text{ \& } (j \in q_a \sim i \in p_a),$$

$$f(b, j, i, b) \Rightarrow y_b(j) = x_b(i) \text{ \& } (j \in q_b \sim i \in p_b),$$

$$a[j,i]a = (s_a, x_a \setminus \{i\}, p_a \setminus \{i\}, y_a \setminus \{j\}, q_a \setminus \{j\}, t_a) \text{ и}$$

$$b[j,i]b = (s_b, x_b \setminus \{i\}, p_b \setminus \{i\}, y_b \setminus \{j\}, q_b \setminus \{j\}, t_b).$$

Пусть $s_a = s_b$ и $x_a \setminus \{i\} = x_b \setminus \{i\}$.

2.1. Докажем выполнение 1-го условия детерминизма:

$$p_a \setminus \{i\} = p_b \setminus \{i\} \text{ и } y_a \setminus \{j\} = y_b \setminus \{j\}.$$

Поскольку $i \neq j$, из $s_a = s_b$ и $x_a \setminus \{i\} = x_b \setminus \{i\}$ следует $y_a(j) = y_b(j)$. А тогда, поскольку $y_a(j) = x_a(i)$ и $y_b(j) = x_b(i)$, имеем $x_a(i) = x_b(i)$. Отсюда, учитывая, что $x_a \setminus \{i\} = x_b \setminus \{i\}$, имеем $x_a = x_b$. Из $s_a = s_b$ и $x_a = x_b$ по 1-ому условию детерминизма для автомата A имеем $y_a = y_b$ и $p_a = p_b$. Итак, $s_a = s_b$, $x_a = x_b$, $p_a = p_b$, $y_a = y_b$. А тогда $p_a \setminus \{i\} = p_b \setminus \{i\}$ и $y_a \setminus \{j\} = y_b \setminus \{j\}$, что и требовалось доказать.

2.2. Пусть также $q_a \setminus \{j\} = q_b \setminus \{j\}$. Докажем выполнение 2-го условия детерминизма: $t_a = t_b$.

По доказанному имеем $s_a = s_b$, $x_a = x_b$, $p_a = p_b$, $y_a = y_b$. Поскольку $j \in q_a \sim i \in p_a$ и $j \in q_b \sim i \in p_b$, а $p_a = p_b$, имеем $j \in q_a \sim j \in q_b$. А поскольку $q_a \setminus \{j\} = q_b \setminus \{j\}$, имеем $q_a = q_b$. Итак, $s_a = s_b$, $x_a = x_b$, $p_a = p_b$, $y_a = y_b$, $q_a = q_b$. А тогда по 2-ому условию детерминизма для автомата A имеем $t_a = t_b$, что и требовалось доказать.

Лемма 2 доказана.

Лемма 3 о контакте внутри «постороннего» автомата: При композиции системы R по соединению $(j,i) \in E$ сохраняется свобода контактов внутри любого автомата C , который не участвует в композиции, т.е. $C \neq \varphi(j)$ и $C \neq \varphi(i)$.

Доказательство: Пусть $(j,i) \in E$, $C \in V$, $C \neq \varphi(j)$, $C \neq \varphi(i)$, $l \in J_C$, $k \in I_C$ и $k \# l$. После композиции $C \in V_{R[j,i]}$, $l \in J_C$, $k \in I_C$ и, поскольку свобода контактов внутри автомата является свойством самого этого автомата независимо от системы, в системе $R[j,i]$ сохраняется свобода контакта $k \#_{R[j,i]} l$.

Лемма 3 доказана.

Лемма 4 о контакте в композиции разных автоматов: При композиции различных детерминированных автоматов по соединению $(j,i) \in E$ сохраняется свобода контакта (k,l) , где $l \neq j$ и $k \neq i$.

Доказательство: Пусть $A = \varphi(j)$, $B = \varphi(i)$ и $A \neq B$. Пусть также автомат $C \in V$, $l \in J_C$, $k \in I_C$, $l \neq j$, $k \neq i$ и $k \# l$. Надо доказать, что $k \#_{R[j,i]} l$. Если $C \neq A$ и $C \neq B$, то по лемме 3 $k \#_{R[j,i]} l$. Рассмотрим остальные случаи (Рис. 2): 1) $C = A \neq B$, 2) $C = B \neq A$. Для этих случаев рассмотрим два перехода в композиции, которые получаются как композиции двух пар переходов a, b и a', b' , где $a, a' \in T_A$ и $b, b' \in T_B$:

$$f(a, j, i, b) \Rightarrow y_a(j) = x_b(i) \text{ \& } (j \in q_a \sim i \in p_b), f(a', j, i, b') \Rightarrow y_a(j) = x_b(i) \text{ \& } (j \in q_a \sim i \in p_b),$$

$$a[j,i]b = (s_a \cup s_b, x_a \cup x_b \setminus \{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b \setminus \{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b) \text{ и}$$

$a[j, i]b = (s_a \cup s_b, x_a \cup x_b / \{i\}, p_a \cup p_b / \{i\}, y_a \cup y_b / \{j\}, q_a \cup q_b / \{j\}, t_a \cup t_b)$

такие, что $s_a \cup s_b = s_a \cup s_b$ и $x_a \cup x_b / \{i\} / \{k\} = x_a \cup x_b / \{i\} / \{k\}$.

Надо доказать, что $(y_a \cup y_b / \{j\})(l) = (y_a \cup y_b / \{j\})(l)$.

$A \neq B$ влечёт $s_a \cup s_b = s_a \cup s_b = \emptyset$. Поэтому $s_a \cup s_b = s_a \cup s_b$ влечёт $s_a = s_a$ и $s_b = s_b$.



1) $C=A \neq B$

2) $C=B \neq A$

Рис. 2. Свободность контакта при композиции разных автоматов

Fig. 2. Free contact with a composition of different machines

1. $C=A \neq B$.

Поскольку $I_A \cap I_B = \emptyset$, $k \in I_A$, $i \in I_B$, имеем $x_a \cup x_b / \{i\} / \{k\} = (x_a / \{k\}) \cup (x_b / \{i\})$.

Аналогично, $x_a \cup x_b / \{i\} / \{k\} = (x_a / \{k\}) \cup (x_b / \{i\})$.

Следовательно, поскольку $x_a \cup x_b / \{i\} / \{k\} = x_a \cup x_b / \{i\} / \{k\}$, имеем $(x_a / \{k\}) \cup (x_b / \{i\}) = (x_a / \{k\}) \cup (x_b / \{i\})$.

Отсюда, поскольку $I_A \cap I_B = \emptyset$, имеем $x_a / \{k\} = x_a / \{k\}$.

Поскольку $J_A \cap J_B = \emptyset$, $l \in J_A$, $j \in J_B$, $l \neq j$, имеем $(y_a \cup y_b / \{j\})(l) = (y_a / \{j\})(l) = y_a(l)$.

Аналогично, $(y_a \cup y_b / \{j\})(l) = y_a(l)$.

Поскольку $k \neq l$, из $s_a = s_a$ и $x_a / \{k\} = x_a / \{k\}$ следует $y_a(l) = y_a(l)$.

Следовательно, $(y_a \cup y_b / \{j\})(l) = (y_a \cup y_b / \{j\})(l)$, что и требовалось доказать.

2. $C=B \neq A$.

Поскольку $I_A \cap I_B = \emptyset$, $k \in I_B$ и $i \in I_B$, имеем $x_a \cup x_b / \{i\} / \{k\} = x_a \cup (x_b / \{i\} / \{k\})$.

Аналогично $x_a \cup x_b / \{i\} / \{k\} = x_a \cup (x_b / \{i\} / \{k\})$.

Следовательно, поскольку $x_a \cup x_b / \{i\} / \{k\} = x_a \cup x_b / \{i\} / \{k\}$, имеем $x_a \cup (x_b / \{i\} / \{k\}) = x_a \cup (x_b / \{i\} / \{k\})$.

Отсюда, поскольку $I_A \cap I_B = \emptyset$, имеем $x_a = x_a$ и $x_b / \{i\} / \{k\} = x_b / \{i\} / \{k\}$.

Поскольку автомат A детерминированный, из $s_a = s_a$ и $x_a = x_a$ следует $y_a = y_a$.

А тогда, поскольку $y_a(j) = x_b(i)$ и $y_a(j) = x_b(i)$, имеем $x_b(i) = x_b(i)$.

Отсюда, поскольку $x_b / \{i\} / \{k\} = x_b / \{i\} / \{k\}$, имеем $x_b / \{k\} = x_b / \{k\}$.

Поскольку $J_A \cap J_B = \emptyset$, $l \in J_B$ и $j \in J_A$, имеем $(y_a \cup y_b / \{j\})(l) = y_b(l)$.

Аналогично $(y_a \cup y_b / \{j\})(l) = y_b(l)$.

Поскольку $k \neq l$, из $s_b = s_b$ и $x_b / \{k\} = x_b / \{k\}$ следует $y_b(l) = y_b(l)$.

Следовательно, $(y_a \cup y_b / \{j\})(l) = (y_a \cup y_b / \{j\})(l)$, что и требовалось доказать.

Лемма 4 доказана.

Будем говорить, что соединение (j,i) смежно с соединением (j',i') , если в графе $\mathbf{Og}(\mathbf{R})$, дуга (j,i) смежна с дугой (j',i') , т.е. вход первого соединения и выход второго соединения образуют контакт (принадлежат одному и тому же автомату): $\varphi(i)=\varphi(j')$. Маршрутом длины m в системе $\mathbf{R}$ назовём ориентированный маршрут длины m в графе $\mathbf{Og}(\mathbf{R})$, т.е. последовательность соединений $Z=(j_0,i_1),(j_1,i_2),\dots,(j_{m-1},i_m)$, в которой каждое соединение, кроме последнего, смежно со следующим соединением: $\forall r=1..m-1 \varphi(i_r)=\varphi(j_r)$. Контакт (i_r,j_r) для $r=1..m-1$ будем называть *контактом в маршруте Z* . Маршрут будем называть *путём*, если в нём все соединения разные: $\forall r,r'=0..m-1 (r \neq r' \Rightarrow (j_r,i_{r+1}) \neq (j_{r'},i_{r'+1}))$. Путь будем называть *циклом*, если $\varphi(i_m)=\varphi(j_0)$, т.е. последнее соединение смежно с первым. В этом случае будем обозначать $i_0=i_m$ и пару $(i_0,j_0)=(i_m,j_0)$ также будем называть *контактом в цикле Z* . Цикл Z будем называть *A-простым*, если $A=\varphi(j_0)$ и $A \neq \varphi(j_r)$ для $r=1..m-1$. Если в цикле Z имеет место $\varphi(j_r)=\varphi(i_{r'})$, то через $Z[j_r,i_{r'}]$ обозначим цикл $(j_r,i_{r+1}),\dots,(j_{r-1},i_r)$, если $r < r'$, и цикл $(j_r,i_{r+1}),\dots,(j_{m-1},i_m),(j_0,i_1),\dots,(j_{r-1},i_r)$, если $r \geq r'$. Цикл будем называть *свободным*, если в цикле есть свободный контакт, т.е. $i_r \neq j_r$ для некоторого $r=0..m-1$.

Лемма 5 о сохранении свободы циклов при композиции:

Если в системе $\mathbf{R}$ все автоматы вполне определённые и детерминированные, все циклы свободные, а $(j,i) \in E$, то в системе $\mathbf{R}[j,i]$ все циклы свободные.

Доказательство: Обозначим компонентные автоматы $A=\varphi(j)$ и $B=\varphi(i)$. Рассмотрим в $\mathbf{R}[j,i]$ цикл $Z=(j_0,i_1),\dots,(j_{m-1},i_m)$. По определению композиции системы $E_{\mathbf{R}[j,i]}=E \setminus \{(j,i)\}$, поэтому все соединения из Z были и в системе $\mathbf{R}$. Также по определению композиции системы $V_{\mathbf{R}[j,i]}=V \setminus \{A,B\} \cup \{A[j,i]B\}$. Смежность соединений из Z может отсутствовать в системе $\mathbf{R}$ только в том случае, когда в $\mathbf{R}[j,i]$ в цикле Z есть контакт (k,l) внутри автомата $A[j,i]B$, но в системе $\mathbf{R}$ вход k и выход l относятся к разным автоматам: 1) $A \neq B$, $k \in I_B$ и $l \in J_A$, или 2) $A \neq B$, $k \in I_A$ и $l \in J_B$. За исключением этого случая Z является циклом в $\mathbf{R}$.

1. Z является циклом в $\mathbf{R}$.

По условию леммы цикл Z в $\mathbf{R}$ свободный. Тогда в $\mathbf{R}$ в цикле Z существует хотя бы один свободный контакт. Покажем, что хотя бы для одного такого контакта $k \neq l$ имеет место $k \neq_{\mathbf{R}[j,i]} l$.

1.1. $A \neq B$.

Пусть $k \neq l$ контакт в $\mathbf{R}$ в цикле Z . Поскольку Z является циклом в $\mathbf{R}[j,i]$, а $(j,i) \notin E_{\mathbf{R}[j,i]}$, имеем $l \neq j$ и $k \neq i$. А тогда по лемме 4 $k \neq_{\mathbf{R}[j,i]} l$.

1.2. $A=B$ и в $\mathbf{R}$ в цикле Z есть контакт $k \neq l$ внутри автомата, отличного от A , т.е. $k \notin I_A$ и, следовательно, $l \notin J_A$.

По лемме 3 $k \neq_{\mathbf{R}[j,i]} l$.

1.3. $A=B$ и в $\mathbf{R}$ в цикле Z любой свободный контакт является контактом внутри автомата A .

Сначала докажем вспомогательное утверждение: В системе $\mathbf{R}$ существует контакт $a \# b$ внутри автомата A в цикле Z , для которого $i \# b$ или $a \# j$.

Так как все циклы в $\mathbf{R}$ свободные, а в Z свободный контакт может быть только внутри автомата A , существует контакт $a \# b$ внутри автомата A в цикле Z .

Если $Z = (j_0, i_1), \dots, (j_{m-1}, i_m)$, то $a = i_r$ и $b = j_r$ для некоторого $r = 0..m-1$. Последовательность $Z' = (b = j_r, i_{r+1}), \dots, (j_{m-1}, i_m)(j_0, i_1), \dots, (j_{r-1}, i_r = a)$, очевидно, тоже является циклом, в котором все соединения и контакты такие же, как в цикле Z .

Цикл Z' можно представить в виде конкатенации A -простых циклов:

$W = Z'[b = l_1, k_2] \cdot Z'[l_2, k_3] \cdot \dots \cdot Z'[l_{w-2}, k_{w-1}] \cdot Z'[l_{w-1}, k_w = a]$ (Рис. 3).

Выберем в W максимальный начальный отрезок $W' = Z'[b = l_1, k_2] \cdot \dots \cdot Z'[l_{w-1}, k_w = c]$ без свободных контактов. В цикле Z для некоторого выхода d существует контакт (c, d) , который по условию максимальной W' свободный.

Рассмотрим цикл $W'' = Z'[b = l_1, k_2] \cdot \dots \cdot Z'[l_{w-1}, k_w = c] \cdot (j, i)$, который получается из W' добавлением соединения-петли (j, i) . В нём нет контактов (a, b) и (c, d) , но есть контакты (i, b) и (c, j) . В цикле W'' все контакты не внутри автомата A не свободные, поскольку эти контакты есть также в цикле Z' и, следовательно, в Z , а в Z все контакты не внутри A не свободные. В цикле W'' внутри A все контакты, кроме, быть может, (i, b) и (c, j) тоже не свободные по построению W' и W'' . Если бы вспомогательное утверждение было не верно, то, поскольку контакты $a \# b$ и $c \# d$, контакты (i, b) и (c, j) были бы не свободные. Но тогда в цикле W'' вообще не было бы свободных контактов, что противоречит тому, что в $\mathbf{R}$ все циклы свободные. Следовательно, вспомогательное утверждение верно.

Теперь докажем основное утверждение пункта 1.3.

Поскольку $A = B$, соединение (j, i) является петлёй в $\mathbf{R}$, и в ней есть единственный контакт (i, j) . По условию леммы петля (как цикл) свободная, следовательно, $i \# j$.

Поскольку $k \# l$, по вспомогательному утверждению $i \# l$ или $k \# j$.

Итак, мы имеем: $k \# l \ \& \ i \# j \ \& \ (i \# l \vee k \# j)$. Докажем, что $k \#_{R[j,i]} l$.

По определению композиции автоматов переход в автомате $A[j, i]A$ получается в результате композиции одного перехода в автомате A . Пусть $a, b \in T_A$ и каждый из переходов a и b компонуется сам с собой, т.е. $f(a, j, i, a) = f(b, j, i, b) = \text{true}$. По определению композиции переходов для $a, b \in T_A$ имеем:

$f(a, j, i, a) \Rightarrow y_a(j) = x_a(i) \ \& \ (j \in q_a \sim i \in p_a)$,

$f(b, j, i, b) \Rightarrow y_b(j) = x_b(i) \ \& \ (j \in q_b \sim i \in p_b)$,

$a[j, i]a = (s_a, x_a/\{i\}, p_a/\{i\}, y_a/\{j\}, q_a/\{j\}, t_a)$ и

$b[j, i]b = (s_b, x_b/\{i\}, p_b/\{i\}, y_b/\{j\}, q_b/\{j\}, t_b)$.

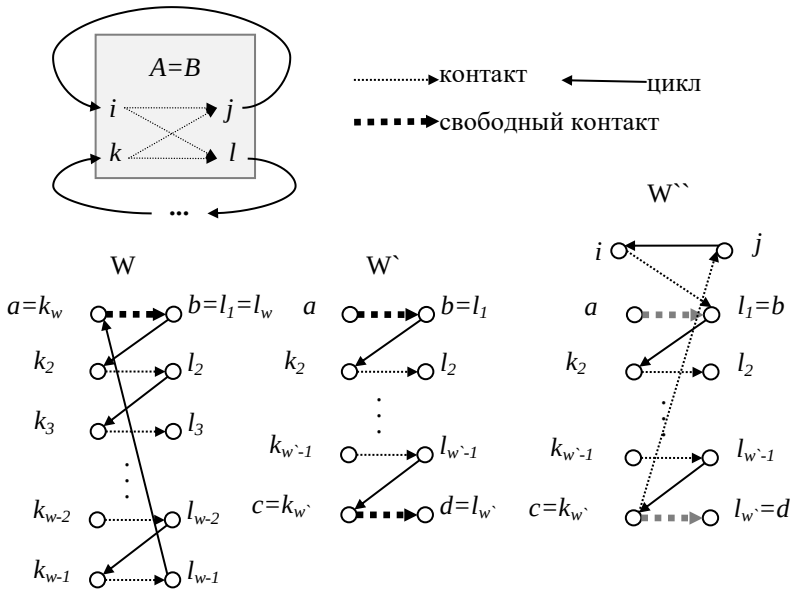


Рис. 3. Свободность цикла при композиции по петле
 Fig. 3. Free cycle when the loop composition

Пусть $s_a = s_b$ и $x_a/\{i\}/\{k\} = x_b/\{i\}/\{k\}$.

Обозначим $s = s_a$, $x = x_a/\{i\}/\{k\}$. Тогда $s = s_b$, $x = x_b/\{i\}/\{k\}$.

Надо доказать, что $(y_a/\{i\})(l) = (y_b/\{i\})(l)$.

Используя обозначения, имеем

$a = (s, x/\{i, x_a(i)\} \cup \{k, x_a(k)\}, p_a, y_a, q_a, t_a)$ и

$b = (s, x/\{i, x_b(i)\} \cup \{k, x_b(k)\}, p_b, y_b, q_b, t_b)$.

Поскольку в автомате определены переходы по всем стимулам, должны быть ещё два перехода:

$c = (s, x/\{i, x_a(i)\} \cup \{k, x_b(k)\}, p_c, y_c, q_c, t_c)$ и

$d = (s, x/\{i, x_b(i)\} \cup \{k, x_a(k)\}, p_d, y_d, q_d, t_d)$.

В переходах a и c стимулы отличаются только по входу k . А тогда, поскольку $k \neq l$, реакции на выходе l одинаковы для этих переходов: $y_a(l) = y_c(l)$. В переходах b и d стимулы тоже отличаются только по входу k . Поэтому, если $k \neq j$, то $y_b(j) = y_d(j)$.

В переходах a и d стимулы отличаются только по входу i . А тогда, поскольку $i \neq j$, реакции на выходе j одинаковы для этих переходов: $y_a(j) = y_d(j)$. В переходах b и c стимулы тоже отличаются только по входу i . Поэтому, если $i \neq l$, то $y_b(l) = y_c(l)$.

Теперь, если $k \neq j$, то $y_a(j) = y_d(j)$ и $y_b(j) = y_d(j)$. Тогда $y_a(j) = y_b(j)$. А поскольку $y_a(j) = x_a(i)$ и $y_b(j) = x_b(i)$, имеем $x_a(i) = x_b(i)$. Но тогда в переходах a и b

стимулы отличаются только по входу k . А поскольку $k \neq l$, имеем $y_a(l) = y_b(l)$. А это влечёт $(y_a/\{i\})(l) = (y_b/\{i\})(l)$.

Если же $i \neq l$, то $y_a(l) = y_c(l)$ и $y_b(l) = y_c(l)$. Тогда $y_a(l) = y_b(l)$. А это влечёт $(y_a/\{i\})(l) = (y_b/\{i\})(l)$.

Следовательно, $k \#_{R[j,i]} l$.

Итак во всех трёх случаях мы доказали, что в R в цикле Z существует хотя бы один свободный контакт (k,l) , который остаётся свободным после композиции $k \#_{R[j,i]} l$. А поскольку контакт (k,l) имеется в цикле Z в $R[j,i]$, цикл Z свободный в $R[j,i]$.

2. Z не является циклом в R , поскольку $A \neq B$, $k \in I_B$ и $l \in J_A$.

Покажем, что любой контакт (k,l) внутри автомата $A[j,i]B$ такой, что $k \in I_B$ и $l \in J_A$, свободный: $k \#_{R[j,i]} l$ (Рис. 4). Очевидно, $k \neq i$ и $l \neq j$.

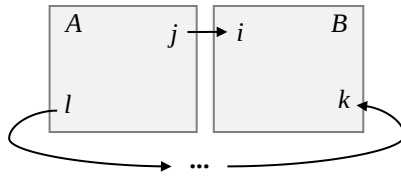


Рис. 4. Свободность контакта (k,l) в $R[j,i]$

Fig. 4. Free contact (k, l) in $R[j, i]$

Рассмотрим два перехода в композиции:

$$f(a, j, i, b) \Rightarrow y_a(j) = x_b(i) \ \& \ (j \in q_a \sim i \in p_b),$$

$$f(a \setminus j, i, b \setminus) \Rightarrow y_a(j) = x_b(i) \ \& \ (j \in q_a \sim i \in p_b),$$

$$a[j, i]b = (s_a \cup s_b, x_a \cup x_b/\{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b/\{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b) \text{ и}$$

$$a \setminus [j, i] b \setminus = (s_a \cup s_b, x_a \cup x_b/\{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b/\{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b)$$

такие, что $s_a \cup s_b = s_a \setminus \cup s_b \setminus$ и $x_a \cup x_b/\{i\}/\{k\} = x_a \setminus \cup x_b \setminus/\{i\}/\{k\}$.

Надо доказать, что $(y_a \cup y_b/\{j\})(l) = (y_a \setminus \cup y_b \setminus/\{j\})(l)$.

Поскольку $A \neq B$, $k \in I_B$ и $i \in I_B$, из $x_a \cup x_b/\{i\}/\{k\} = x_a \setminus \cup x_b \setminus/\{i\}/\{k\}$ следует $x_a = x_a \setminus$.

А тогда, поскольку автомат A детерминированный, $y_a(l) = y_a \setminus(l)$, что вместе с $j \in J_A$ и $l \in J_A$, влечёт $(y_a \cup y_b/\{j\})(l) = (y_a \setminus \cup y_b \setminus/\{j\})(l)$.

Следовательно, $k \#_{R[j,i]} l$, а поскольку контакт (k,l) имеется в цикле Z в $R[j,i]$, цикл Z свободный в $R[j,i]$.

3. Z не является циклом в R , поскольку $A \neq B$, $k \in I_A$ и $l \in J_B$.

Очевидно, $k \neq i$ и $l \neq j$. Контакт (a,b) внутри автомата $A[j,i]B$ будем называть *AB-разрывом*, если $a \in I_A$ и $b \in J_B$, и *BA-разрывом*, если $a \in I_B$ и $b \in J_A$.

Цикл Z можно представить как конкатенацию путей без разрывов в каждом пути и с разрывами между путями: $Z = Z_1 \cdot Z_2 \dots Z_w$. Для $v = 1..w$ через k_v обозначим вход последнего соединения в Z_v , а через l_v обозначим выход первого соединения в Z_v .

Тогда для $v=1..w-1$ должно быть $\varphi_{R[j,i]}(k_v)=\varphi_{R[j,i]}(l_{v+1})=A[j,i]B$ и контакт (k_v, l_{v+1}) является разрывом, и $\varphi_{R[j,i]}(k_w)=\varphi_{R[j,i]}(l_1)=A[j,i]B$ и контакт (k_w, l_1) является разрывом. Так как в каждом пути Z_v нет разрывов, он является путём в R .

Если в $R[j,i]$ в цикле Z есть BA -разрыв (a,b) , то по доказанному в п.2 $a \#_{R[j,i]} b$, следовательно, цикл Z свободный в $R[j,i]$.

Если в каком-то из путей Z_v , где $v=1..w$, есть контакт $a \# b$, то, поскольку путь Z_v является также путём в $R[j,i]$, должно быть $a \neq i$ и $b \neq j$. А тогда по лемме 4 $a \#_{R[j,i]} b$, следовательно, цикл Z свободный в $R[j,i]$.

Далее будем считать, что в $R[j,i]$ в цикле Z есть только AB -разрывы, и в каждом из путей Z_v , где $v=1..w$, нет свободных контактов (Рис. 5).

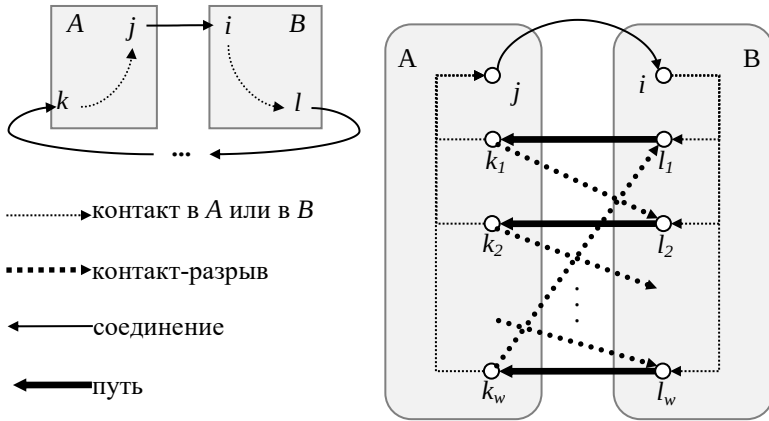


Рис. 5. Цикл как конкатенация путей с разрывами между путями

Fig. 5. The cycle as a concatenation of paths with breaks between paths

Для каждого $v=1..w$ путь Z_v вместе с соединением (j,i) образует в R цикл $Z_v = Z_v \cdot (j,i)$. В цикле Z_v есть два контакта внутри автоматов A и B: контакт (k_v, j) в автомате A и контакт (i, l_v) в автомате B. Поскольку остальные контакты цикла Z_v являются контактами пути Z_v , они не свободные. Поскольку все циклы в R свободные, цикл Z_v должен быть свободным, а тогда должно быть $k_v \# j$ или $i \# l_v$.

Нам надо показать, что для некоторого v имеет место $k_v \#_{R[j,i]} l_{v+1}$, если $v=1..w-1$, или $k_w \#_{R[j,i]} l_1$, если $v=w$.

Допустим для некоторого $v=1..w$ имеет место $k_v \# j$.

Обозначим $k=k_v$ и $l=l_{v+1}$, если $v=1..w-1$, или $l=l_1$, если $v=w$. Имеем $k \# j$. Надо показать, что $k \#_{R[j,i]} l$.

Рассмотрим два перехода в композиции:

$$f(a, j, i, b) \Rightarrow y_a(j) = x_b(i) \ \& \ (j \in q_a \sim i \in p_b),$$

$$f(a', j, i, b') \Rightarrow y_{a'}(j) = x_{b'}(i) \ \& \ (j \in q_{a'} \sim i \in p_{b'}),$$

$a[j,i]b = (s_a \cup s_b, x_a \cup x_b / \{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b / \{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b)$ и
 $a'[j,i]b' = (s_a' \cup s_b', x_a' \cup x_b' / \{i\}, p_a' \cup p_b' \setminus \{i\}, y_a' \cup y_b' / \{j\}, q_a' \cup q_b' \setminus \{j\}, t_a' \cup t_b')$
 такие, что $s_a \cup s_b = s_a' \cup s_b'$ и $x_a \cup x_b / \{i\} / \{k\} = x_a' \cup x_b' / \{i\} / \{k\}$.

Достаточно показать, что $(y_a \cup y_b / \{j\})(l) = (y_a' \cup y_b' / \{j\})(l)$.

Поскольку $A \neq B$, $k \in I_A$ и $i \in I_B$, из $x_a \cup x_b / \{i\} / \{k\} = x_a' \cup x_b' / \{i\} / \{k\}$ следует $x_a / \{k\} = x_a' / \{k\}$ и $x_b / \{i\} = x_b' / \{i\}$.

Поскольку $k \neq j$, из $x_a / \{k\} = x_a' / \{k\}$ следует $y_a(j) = y_a'(j)$, что влечёт $x_b(i) = x_b'(i)$. А это вместе с $x_b / \{i\} = x_b' / \{i\}$ влечёт $x_b = x_b'$. Но тогда, поскольку автомат B детерминированный, $y_b = y_b'$, что влечёт $(y_a \cup y_b / \{j\})(l) = (y_a' \cup y_b' / \{j\})(l)$, поскольку $A \neq B$, $j \in J_A$ и $l \in J_B$.

Теперь предположим, что для каждого $v = 1..w$ не верно, что $k_v \neq j$.

Тогда для каждого $v = 1..w$ имеет место $i \neq l_v$.

Выберем любое $v = 1..w$ и обозначим $l = l_v$ и $k = k_{v-1}$, если $v = 2..w$, или $k = k_w$, если $v = 1$. Имеем $i \neq l$. Надо показать, что $k \neq_{R[j,i]} l$.

Рассмотрим два перехода в композиции:

$f(a, j, i, b) \Rightarrow y_a(j) = x_b(i) \text{ \& } (j \in q_a \sim i \in p_b)$,

$f(a', j, i, b') \Rightarrow y_a'(j) = x_b'(i) \text{ \& } (j \in q_a' \sim i \in p_b')$,

$a[j,i]b = (s_a \cup s_b, x_a \cup x_b / \{i\}, p_a \cup p_b \setminus \{i\}, y_a \cup y_b / \{j\}, q_a \cup q_b \setminus \{j\}, t_a \cup t_b)$ и

$a'[j,i]b' = (s_a' \cup s_b', x_a' \cup x_b' / \{i\}, p_a' \cup p_b' \setminus \{i\}, y_a' \cup y_b' / \{j\}, q_a' \cup q_b' \setminus \{j\}, t_a' \cup t_b')$

такие, что $s_a \cup s_b = s_a' \cup s_b'$ и $x_a \cup x_b / \{i\} / \{k\} = x_a' \cup x_b' / \{i\} / \{k\}$.

Достаточно показать, что $(y_a \cup y_b / \{j\})(l) = (y_a' \cup y_b' / \{j\})(l)$.

Поскольку $A \neq B$, $k \in I_A$ и $i \in I_B$, из $x_a \cup x_b / \{i\} / \{k\} = x_a' \cup x_b' / \{i\} / \{k\}$ следует $x_b / \{i\} = x_b' / \{i\}$.

Поскольку $i \neq l$, из $x_b / \{i\} = x_b' / \{i\}$ следует $y_b(l) = y_b'(l)$, что влечёт $(y_a \cup y_b / \{j\})(l) = (y_a' \cup y_b' / \{j\})(l)$, поскольку $A \neq B$, $j \in J_A$ и $l \in J_B$.

Лемма 5 доказана.

Теорема 1 о вполне определенности и детерминированности системы автоматов:

Если в системе $R = (M, V, E)$ все автоматы вполне определенные и детерминированные, и все циклы соединений свободные, то система $R[Z^\wedge]$, где $Z^\wedge$ – последовательность всех соединений, состоит из вполне определенных детерминированных автоматов, не связанных друг с другом. При этом система $R[Z^\wedge]$ не зависит от выбора последовательности $Z^\wedge$ для одного и того же множества E .

Доказательство: Так как в системе $R = (M, V, E)$ все автоматы вполне определенные и детерминированные и все соединения-петли (как циклы соединений) свободные, по леммам 1 и 2 в композиции $R[j,i]$ для любого соединения $(j,i) \in E$ все автоматы тоже вполне определенные и детерминированные. Так как в системе R все автоматы вполне определенные и детерминированные, и все циклы свободные, по лемме 5 в композиции $R[j,i]$

все циклы тоже свободные. Таким образом, в системе $R[j,i]$, как и в исходной системе R , все автоматы вполне определенные и детерминированные, и все циклы соединений свободные. Будем выполнять композицию до тех пор, пока в системе есть соединения. Так как число соединений в системе конечно, а при композиции оно уменьшается на 1, через конечное число композиций (равное числу соединений в R) получим систему $R[Z^*]$, в которой нет соединений, и которая состоит только из вполне определенных и детерминированных автоматов. По ассоциативности композиции (теорема 3 в [2]) система $R[Z^*]$ не зависит от выбора последовательности Z^* для одного и того же множества E .

Теорема 1 доказана.

4. Системы автоматов класса «вершины-дуги»

В нашей статье [1] система автоматов описывалась как граф, в вершинах которого находились автоматы, а дуги соответствовали каналам связи, через которые автоматы обменивались сообщениями. При этом дуга представляла собой очередь сообщений длины 1. Она моделировалась автоматом с одним входом и одним выходом. Переход в автомате (вершины или дуги) помечался только парой (x,y) , где x – стимул, а y – реакция, в отличие от более общих (x,p,y,q) -переходов в модели, используемой в данной статье. Автоматы вершин и дуг были детерминированными, но, кроме того, в автомате дуги реакция (сообщение, выбираемое из начала очереди) зависела только от состояния автомата, т.е. не зависела от стимула (сообщения, помещаемого в конец очереди).

Для того чтобы интерпретировать автоматы из [1] в терминах данной статьи, выполним следующие преобразования:

1. Нам нужно заменить стимул x и реакцию y как частичные отображения на полные отображения x' и y' , а также добавить параметры p и q . В каждом автомате каждый переход (s,x,y,t) заменяется на множество кратных переходов вида (s,x',p,y',q,t) , где стимулы x и x' совпадают на домене x , т.е. $\forall i \in \text{Dom}(x) x(i)=x'(i)$, $p = \text{Dom}(x)$, реакции y и y' совпадают на домене y , а на выходах из его дополнения сообщения пустые, т.е. $\forall j \in \text{Dom}(y) y(j)=y'(j)$ и $\forall j \notin \text{Dom}(y) y'(j)=\Lambda$, $q = \text{Dom}(y)$.
2. Если после этого в автомате в некотором состоянии s нет перехода по некоторому стимулу x' , то добавляется переход-петля по этому стимулу без приёма и выдачи сообщений: $(s,x',\emptyset, Jx\{\Lambda\},\emptyset,s)$.
3. Кроме этого, в автомате дуги (очередь длины 1) с входом i и выходом j добавляются переходы для любого стимула без приёма сообщений для случая, когда на дуге находится сообщение, которое не может быть передано: $\forall m \in M \forall m' \in M_A (m,(i,m'),\emptyset,(j,m),\emptyset,m)$.

Легко видеть, что после такого преобразования автомат вершины определен по всем стимулам, а автомат дуги вполне определен. Если автоматы

детерминированы в смысле [1], то после преобразования они детерминированы в терминах данной статьи. Также в автомате дуги реакция зависит только от состояния (не зависит от стимула).

Однако для моделирования системы автоматов из [1] в терминах данной статьи мы не можем применять композицию, которая определена в данной статье (и в [2]). Дело в том, что в [1] предполагалась *семантика с асинхронными переходами*, когда один автомат «стоит» (не выполняет никакого перехода), а связанный с ним автомат может работать (выполнять переход). Это происходит тогда, когда автоматы некоторых дуг, выходящих из вершины, не принимают сообщения, посылаемые на эти дуги автоматом этой вершины. В этом случае автомат вершины «стоит», а автоматы выходящих дуг могут выполнять переходы, передавая находящиеся на них сообщения в автоматы вершины, в которые эти дуги входят. В данной статье (и в [2]), как указано в начале раздела 3, используется *семантика без асинхронных переходов*: переход в каждом автомате выполняется одновременно с переходами в автоматах, связанных с ним по соединениям, поэтому, когда один автомат «стоит», все связанные с ним автоматы тоже «стоят».

Для моделирования семантики с асинхронными переходами мы предложим специальную асинхронную композицию автомата вершины с автоматами всех выходящих дуг. Почему автомат вершины следует единовременно компоновать с автоматами сразу всех выходящих дуг? Причина этого в том, что в семантике статьи [1] автомат вершины выполняет переход, принимая сообщения со своих входов, только в том случае, когда удастся передать по его выходам все непустые сообщения, описываемые реакцией перехода. Иными словами, в терминах (x, p, u, q) -переходов параметр q всегда был равен множеству тех выходов j , по которым передавалось непустое сообщение $u(j) \neq \Lambda$. Поэтому то, «стоит» автомат вершины или выполняет переход, зависит от состояния автоматов всех выходящих дуг.

Мы также обобщим понятие дуги так, чтобы автомат дуги мог моделировать не только очередь длины 1, но также очередь любой длины, очередь с приоритетами сообщений, стек и т.п. В самом общем виде дуга будет рассматриваться как автомат-посредник между автоматами вершин, но с одним важным ограничением: у автомата дуги должен быть только один вход. Заметим, что выходов в автомате дуги может быть несколько.

Будем говорить, что система автоматов $R = (M, V, E)$ относится к классу «вершины-дуги», если 1) множество V автоматов разбито на два подмножества: автоматы вершин и автоматы дуг; 2) автоматы вершин и дуг чередуются: каждое соединение ведёт либо из автомата вершины в автомат дуги, либо из автомата дуги в автомат вершины; 3) автоматы вершин определены по всем стимулам, а автоматы дуг вполне определены; 4) все автоматы детерминированные; 5) в автомате дуги реакция зависит только от состояния, т.е. не зависит от стимула, 6) автомат дуги имеет только один вход.

Асинхронная композиция автомата вершины с автоматами всех выходящих дуг.

Пусть в системе $R = (M, V, E)$ класса «вершины-дуги» имеется автомат вершины A и автоматы выходящих дуг $B_1, \dots, B_n$ так, что каждое соединение (j, i) , где $j \in J_A$, ведёт в некоторый из автоматов дуг, т.е. $\exists m=1..n$ такое, что $i \in I_{B_m}$, и для каждого $m=1..n$ найдётся такое соединение (l, k) , что $l \in J_A$ и $k \in I_{B_m}$.

Для m -ого ($m=1..n$) автомата дуги множество его входов, используемых в соединениях с автоматом вершины, равно $I_m = \{i \in I_{B_m} | \exists j \in J_A (j, i) \in E\}$. Множество I входов композиции определим как множество всех входов автоматов-операндов, кроме тех, что используются в соединениях автомата вершины с автоматами дуг: $I = I_A \cup (I_{B_1} \setminus I_1) \cup \dots \cup (I_{B_n} \setminus I_n)$. Множество J выходов композиции определим как множество всех выходов автоматов дуг: $J = J_{B_1} \cup \dots \cup J_{B_n}$. Состояние композиции – это набор состояний автоматов-операндов $s = (s_A, s_1, \dots, s_n) \in S_A \times S_{B_1} \times \dots \times S_{B_n}$. Стимул композиции $x: I \rightarrow M_A$.

Для данного композиционного состояния s и композиционного стимула x в автомате вершины выделяется его состояние s_A и его стимул $x_A = x \cap (I_A \times M_A)$. Поскольку автомат вершины определен по всем стимулам и детерминирован, пара (s_A, x_A) однозначно определяет множество переходов вида $a = (s_A, x_A, p_A, y_A, q_A, t_A) \in T_A$, отличающихся друг от друга только параметром q_A и постсостоянием t_A , однозначно определяемым остальными параметрами перехода. Реакция y_A вместе с композиционным стимулом x однозначно определяют стимул для каждого m -го автомата дуги $x_m = (x \cap (I_{B_m} \times M_A)) \cup \{(i, y_A(j)) | i \in I_m \text{ \& } (j, i) \in E\}$. Поскольку автомат дуги вполне определен и детерминирован, пара (s_m, x_m) однозначно определяет в автомате дуги множество переходов вида $a_m = (s_m, x_m, p_m, y_m, q_m, t_m) \in T_{B_m}$, отличающихся друг от друга только параметром q_m , который пробегает все возможные значения, и постсостоянием t_m , однозначно определяемым остальными параметрами перехода. Поскольку для каждого $(j, i) \in E$, где $i \in I_m$, имеет место $x_m(i) = y_A(j)$, переход a выполняется, если для каждого $m=1..n$ выполняется оставшая часть условия композиции перехода a с переходом a_m по соединению $(j, i): j \in q_A \sim i \in p_m$.

Если эти условия выполнены для некоторого перехода a , т.е. для некоторого q_A , то они, очевидно, не выполнены для остальных q_A . Тогда определяется композиционный переход (s, x, p, y, q, t) , где $p = p_A \cup (p_1 \setminus I_1) \cup \dots \cup (p_n \setminus I_n)$, $y = y_1 \cup \dots \cup y_n$, $q = q_1 \cup \dots \cup q_n$, $t = (t_A, t_1, \dots, t_n)$.

Если эти условия не выполнены ни для какого перехода a , т.е. ни для какого q_A , то автомат вершины не меняет состояние, не принимает и не передает сообщения, поэтому на входах автоматов дуг, соединенных с выходами автомата вершины, находятся пустые сообщения. Эти пустые сообщения вместе с композиционным стимулом x однозначно определяют стимул для каждого m -го автомата дуги $x'_m = (x \cap (I_{B_m} \times M_A)) \cup \{(i, \Lambda) | i \in I_m\}$. Поскольку автомат дуги вполне определен и детерминирован, пара (s_m, x'_m) однозначно

определяет в автомате дуги множество переходов вида $a_m = (s_m, x_m, p_m, y_m, q_m, t_m)$, отличающихся друг от друга только параметром q_m , который пробегает все возможные значения, и постсостоянием t_m , однозначно определяемым остальными параметрами перехода. В этом случае определяется композиционный переход (s, x, p, y, q, t) , где $p = (p_1 | I_1) \cup \dots \cup (p_n | I_n)$, $y = y_1 \cup \dots \cup y_n$, $q = q_1 \cup \dots \cup q_n$, $t = (s_A, t_1, \dots, t_n)$. Этот композиционный переход как раз и моделирует асинхронные переходы в автоматах дуг, когда автомат вершины «стоит».

Теорема 2 об асинхронной композиции автомата вершины с автоматами всех выходящих дуг:

В системе автоматов класса «вершины-дуги» асинхронная композиция автомата вершины с автоматами всех выходящих дуг вполне определена и детерминирована, а все её контакты свободные.

Доказательство: Определенность композиции по всем стимулам следует из определения композиции. Определенность композиции по всем q следует, кроме того, из вполне определенности автоматов дуг. Детерминированность композиции следует из определения композиции и детерминированности автоматов вершин и дуг. Поскольку реакция в автомате дуги зависит только от состояния дуги, а в композиции реакция состоит из реакций автоматов дуг, композиционная реакция зависит только от композиционного состояния, содержащего все состояния автоматов дуг. Тем самым, все контакты в композиции свободные.

Теорема 2 доказана.

Поскольку в системе класса «вершины-дуги» автоматы вершин и дуг строго чередуются, и каждая дуга имеет только один вход, асинхронную композицию можно проводить до тех пор, пока остаются автоматы вершин. В конечном счете останутся только вполне определенные и детерминированные автоматы, полученные с помощью асинхронной композиции, все контакты которых свободные, а также автоматы внешних входных дуг, которые как и любые автоматы дуг, тоже вполне определенные и детерминированные, а все их контакты свободные. После этого мы уже можем использовать семантику без асинхронных переходов, применяя композицию, определенную в данной статье (и в [2]). После завершения такой композиции системы по теореме 1 гарантируется детерминизм системы класса «вершины-дуги».

Возникает вопрос: а что делать, если дуга имеет несколько входов? Возможно ли соответствующее обобщение системы автоматов класса «вершины-дуги» в семантике с асинхронными переходами? Как было сказано выше, асинхронная композиция должна компоновать автомат вершины с автоматами сразу всех выходящих из вершины дуг. Поэтому, если дуга имеет несколько входов, то её следует компоновать сразу со всеми вершинами, выходы которых соединены со входами этой дуги. В целом в асинхронной композиции должен участвовать двудольный подграф системы: все соединения в нем ведут из вершин в дуги, и подграф замкнут по выходящим

из вершин соединениям. Тем самым в асинхронной композиции может участвовать сразу несколько автоматов вершин, и некоторые из них могут «стоять».

В целом получается следующая процедура. Поскольку в автомате дуги реакция зависит только от её состояния, состояния автоматов всех дуг однозначно определяют сообщения на выходах автоматов всех дуг. Поскольку автоматы вершин и дуг чередуются, тем самым однозначно определены сообщения на входах автоматов всех вершин. Поскольку автомат вершины определен по всем стимулам и детерминирован, в каждом автомате вершины однозначно определяется параметр p и реакция u подмножества выполнимых переходов. Эти реакции однозначно определяют сообщения на выходах автоматов всех вершин. Поскольку автоматы вершин и дуг чередуются, тем самым однозначно определены сообщения на входах автоматов всех дуг. Поскольку автомат дуги определен по всем стимулам и детерминирован, в каждом автомате дуги однозначно определяется параметр p подмножества выполнимых переходов.

После этого по каждому соединению (j,i) , ведущему из автомата вершины $\varphi(j)$ в автомат дуги $\varphi(i)$, проверяется согласованность параметров p и q , т.е. проверяется, что $j \in q \sim i \in p$. Если это не так, автомат вершины $\varphi(j)$ должен «стоять», но тогда меняются сообщения на его выходах и, соответственно, сообщения на входах дуг, с которыми эти выходы соединены: эти сообщения становятся пустыми. В этом случае требуется повторно «пересчитать» подмножества выполнимых переходов в автоматах дуг.

Если проверка согласованности параметров p и q по соединениям и «пересчет» подмножеств выполнимых переходов в автоматах дуг будут чередоваться, то это может привести к недетерминизму, т.е. результирующее состояние системы (как набор состояний всех автоматов) будет зависеть от того, в каком порядке проверяются соединения (пример на Рис. 6). Поэтому нужно сначала выполнить проверку всех соединений, определяя, какие автоматы вершин будут «стоять», а уже потом заново «пересчитывать» подмножества выполнимых переходов в автоматах дуг. Этот процесс может потребовать несколько итераций, но он гарантированно «сходится», поскольку на каждом шаге, кроме последнего, число «стоящих» автоматов вершин увеличивается, а общее число вершин конечно.

Получившаяся процедура асинхронной композиции достаточно сложная, а её семантика недостаточно «прозрачна». Это наводит на мысль попробовать разработать новую модель взаимодействия автоматов, которая совмещала бы в себе достоинства семантики с асинхронными переходами и семантики без асинхронных переходов.

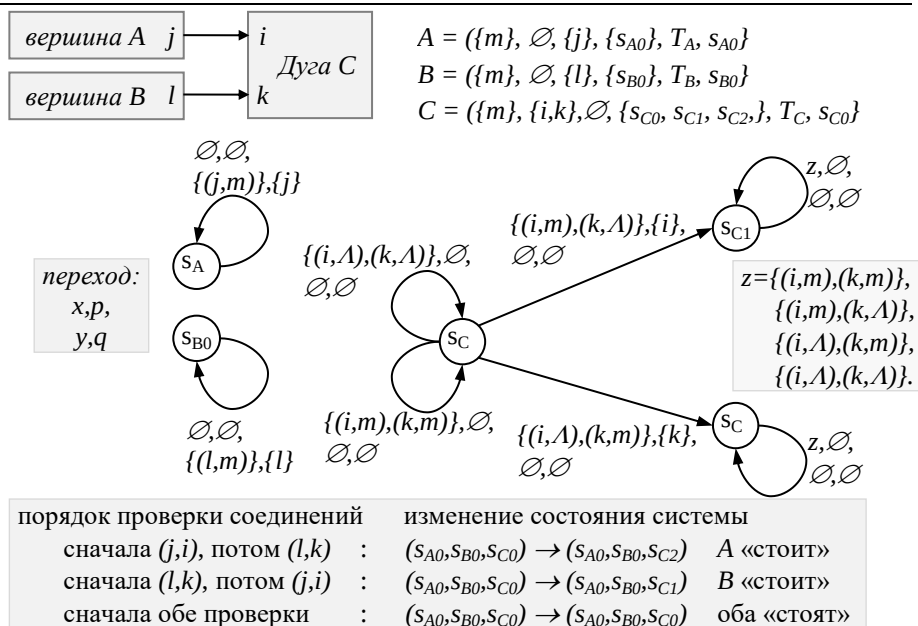


Рис. 6. Пример недетерминизма
Fig. 6. An example of nondeterminism

5. Тестирование

Если для системы $R=(M,V,E)$ не делается никаких предположений о том, правильно ли работают автоматы системы из множества V и правильно ли (не меняющееся) множество соединений E , то для тестирования системы тесты генерируются по композиции $R[Z^{\wedge}]$. Если композиция состоит из нескольких автоматов, то, поскольку эти автоматы не связаны между собой, то тесты можно генерировать для каждого автомата из $V_{R[Z^{\wedge}]}$ независимо: полный набор тестов системы совпадает с объединением полных наборов тестов для всех автоматов из $V_{R[Z^{\wedge}]}$. Иными словами, каждый компонент связности системы R тестируется отдельно и независимо от других компонентов связности, что естественно.

Как указывалось в нашей предыдущей статье [1], если в соединениях (множестве E) нет ошибок (будем называть это *гипотезой о связях*), то есть множество соединений совпадает с заданным, то такое тестирование системы сводится к проверке правильности переходов каждого автомата из V_R . Поскольку компонент тестируется как часть системы, не все переходы автомата компонента, которые можно проверить при автономном тестировании с прямым доступом к компоненту, можно проверить при тестировании системы. Поэтому речь идёт только о *достижимых переходах*

автоматов, то есть таких переходах, которые могут быть выполнены при работе автомата как части системы. С другой стороны, если нас интересует правильность работы системы в целом, то достаточно «правильности» только достижимых переходов.

Тест моделируется автоматом, выходы которого соединяются с внешними входами системы, а внешние выходы системы соединяются со входами теста. В результате получается замкнутая система, т.е. система без внешних входов и выходов. Автоматы такой системы, в том числе тест, работают синхронно: на каждом такте каждый автомат выполняет один переход.

В данном разделе предлагается алгоритм построения набора тестов, который является полным (проверяет все переходы автоматов компонентов, достижимые при работе этих компонентов в системе) при выполнении двух условий: 1) верна гипотеза о связях, 2) система детерминирована. Дополнительно этот алгоритм определяет недостижимые переходы автоматов компонентов. Предполагается, во-первых, что нам известно, каким должен быть автомат каждого компонента (задан граф переходов автомата с точностью до изоморфизма) и именно это должно проверяться при тестировании. Во-вторых, тестовая система может не только выполнять тест, посылая сообщения на внешние входы системы и принимая сообщения с внешних выходов системы, но и наблюдать как состояния автоматов системы, так и сообщения, передаваемые по соединениям между автоматами системы.

Поскольку мы не налагаем никаких ограничений на связность графов переходов автоматов, вообще говоря, полный набор тестов содержит более одного теста. Это означает, что требуется рестарт системы при переходе от одного теста к другому: при рестарте происходит «подмена» одного теста другим.

Для решения этой задачи мы предлагаем использовать любой алгоритм генерации полного набора тестов для одного детерминированного автомата. Таких алгоритмов предложено довольно много, они базируются на построении набора маршрутов, покрывающих граф переходов автомата, достижимых из его начального состояния (смотри, например, [8]). В качестве такого автомата для наших целей выбирается каждый из автоматов композиционной системы $R[Z/\mathcal{N}]$. Понятно, что покрывая все достижимые переходы автомата $A^\wedge \in V_{R[Z/\mathcal{N}]}$, мы покрываем все достижимые переходы всех автоматов из соответствующего компонента связности системы R , который обозначим $V(A^\wedge)$. Однако такой набор тестов может оказаться сильно избыточным для решения нашей задачи: покрытие всех достижимых переходов всех автоматов из $V(A^\wedge)$ не обязательно требует покрытия всех достижимых переходов автомата $A^\wedge$.

Поэтому предлагается в процессе генерации полного набора тестов для автомата $A^\wedge$ применять процедуру фильтрации, которая будет отбрасывать «лишние» тесты. Эта процедура работает следующим образом. С самого начала создаётся пустое множество T генерируемого набора тестов и

множество P непокрытых переходов автоматов из $V(A^\wedge)$, которое сначала равно множеству всех переходов всех автоматов из $V(A^\wedge)$. Когда генерируется очередной i -ый тест T_i для автомата $A^\wedge$, вычисляется множество P_i переходов автоматов из $V(A^\wedge)$, покрываемое этим тестом. Тесту соответствует маршрут $Z^\wedge$ в автомате $A^\wedge$. Каждому переходу маршрута $Z^\wedge$ соответствует множество переходов всех автоматов из $V(A^\wedge)$: по одному переходу на каждый автомат из $V(A^\wedge)$. Объединение этих множеств для всех переходов из $Z^\wedge$ и есть множество P_i . Далее алгоритм фильтрации проверяет, покрывает ли i -ый тест какой-либо новый, ещё не покрытый переход какого-либо автомата компонента. Если $P_i \cap P = \emptyset$, то никаких новых переходов i -ый тест не покрывает, и он отбрасывается. В противном случае тест добавляется к набору тестов $T := T \cup \{T_i\}$, а из множества непокрытых переходов удаляются новые переходы $P := P \setminus P_i$. После того как все тесты сгенерированы и отфильтрованы, получившееся множество T является искомым полным набором тестов, а множество P – множеством недостижимых переходов автоматов компонентов.

6. Заключение

В статье [1] в заключении были перечислены направления дальнейших исследований из семи пунктов. После статьи [2] и данной статьи остаются четыре пункта, которые мы здесь и перечислим.

1) Предложенный алгоритм фильтрации не обязательно даёт оптимальный полный набор тестов. Оптимальность может пониматься как минимальное число или минимальная суммарная длина тестовых последовательностей. Соответственно, возникает задача оптимизации, т.е. поиска оптимального набора, которая, вообще говоря, сводима к задаче о поиске минимального покрытия ([9], [10]).

2) При тестировании, как обычно, возникает задача «огрубления», когда проверяется не «всё подряд», а проверка делается выборочно. Это называют факторизацией тестирования [11]. Например, для проверки правильности реализации числовой функции домены её аргументов разбиваются на поддомены, и из каждого поддомена выбирается хотя бы одно число для проверки. В терминах рассматриваемой здесь задачи факторизация означает, что на переходах автомата компонента определяется отношение эквивалентности, и считается, что достаточно проверить хотя бы один переход из класса эквивалентности. Задача заключается в том, чтобы уметь находить «правильные» отношения эквивалентности, т.е. отношения, удовлетворяющие некоторым практическим потребностям.

6) В данной статье предлагается решение проблемы тестирования компонентов только детерминированной системы автоматов. Одним из направлений дальнейших исследований могло бы стать исследование проблемы недетерминизма. Более точно, ставится задача определить такие ограничения на недетерминизм системы и/или составляющих её автоматов,

которые позволяли бы выполнять полное тестирование за конечное время. Для автономного тестирования, когда автомат находится под непосредственным управлением теста (не в контексте окружающей его части системы), предложенные неплохие решения этой задачи ([12], [13], [14], [15]).

7) В данной статье предполагается, что известно, каким должен быть автомат каждого компонента: задан граф переходов автомата с точностью до изоморфизма. В более общем случае между автоматом компонента в реализации и автоматом компонента в спецификации задаётся то или иное отношение конформности, которое слабее изоморфизма: квази-редукция, симуляция и т.п. Это требует более сложного алгоритма тестирования. Если тестирование компонента автономное, т.е. не в контексте других компонентов системы и связующих их дуг, то возможно полное тестирование за конечное время конформности типа редукции или слабой симуляции ([4], [5], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]). Для составной системы возникает проблема декомпозиции системных требований, называемая также проблемой несохранения конформности. Она заключается в том, что композиция реализаций компонентов, конформных спецификациям этих компонентов, в общем случае неконформна спецификации системы, в частности, композиции спецификаций компонентов. Этой проблеме посвящён ряд работ ([4], [6], [23]), но остаётся задача переосмысления предложенных решений для рассматриваемой в этой статье задачи тестирования компонентов системы при условии, что верна гипотеза о связях.

Список литературы

- [1]. И.Б.Бурдонов, А.С.Косачев. Тестирование системы автоматов. Труды ИСП РАН, том 28(1), 2016 г. DOI: 10.15514/ISPRAS-2016-28(1)-7
- [2]. И.Б.Бурдонов, А.С.Косачев. Система автоматов: композиция по графу связей. Труды ИСП РАН, том 28(1), 2016 г. DOI: 10.15514/ISPRAS-2016-28(1)-8
- [3]. Revised Working Draft on “Framework: Formal Methods in Conformance Testing”. JTC1/SC21/WG1/Project 54/1, ISO Interim Meeting, ITU-T on, Paris, 1995 г.
- [4]. И.Б.Бурдонов, Косачев А.С., В.В.Кулямин. Теория соответствия для систем с блокировками и разрушением. «Физ-мат лит» Наука, Москва, 2008 г., 412 стр.
- [5]. И.Б.Бурдонов. Теория конформности (функциональное тестирование программных систем на основе формальных моделей). LAP Lambert Academic Publishing, 2011 г., 428 стр.
- [6]. И.Б.Бурдонов, А.С.Косачев. Пополнение спецификации для ioco. Программирование, 2011 г., №1, стр. 3-18.
- [7]. А. Камкин, М. Чупилко. Обзор современных технологий имитационной верификации аппаратуры. Программирование, 2011 г., №3, стр. 42-49.
- [8]. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин. Неизбыточные алгоритмы обхода ориентированных графов. Детерминированный случай. Программирование, 2003 г., №5, стр. 59-69.
- [9]. Ананий В. Левитин. Алгоритмы: введение в разработку и анализ. М.: «Вильямс», 2006 г., стр. 160-163, ISBN 0-201-74395-7.

- [10]. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. 2-ое издание. М.: «Вильямс», 2006 г., стр. 456-458, ISBN 0-07-013151-1.
- [11]. Бурдонов И.Б., Косачев А.С., Кулямин В.В. Использование конечных автоматов для тестирования программ. Программирование. 2000 г., № 2, стр.12-28.
- [12]. И.Б. Бурдонов, А.С. Косачев. Полное тестирование с открытым состоянием ограниченно недетерминированных систем. Программирование, 2009 г., №6, стр. 3-18.
- [13]. И.Б.Бурдонов, А.С.Косачев. Семантики взаимодействия с отказами, дивергенцией и разрушением. Часть 2. Условия конечного полного тестирования. Вестник Томского Государственного Университета, № 2(15), 2011 г., стр. 89-98.
- [14]. И.Б. Бурдонов, А.С. Косачев. Тестирование конформности на основе соответствия состояний. Труды ИСП РАН, № 18, 2010 г., стр. 183-220.
- [15]. И.Б.Бурдонов, А.С.Косачев, Безопасное тестирование симуляции систем с отказами и разрушением. Моделирование и анализ информационных систем, том 17(4), 2010 г., стр. 27-40.
- [16]. I.B.Burdonov, A.S.Kossatchev, V.V.Kuliamin. Formal Conformance Testing of Systems with Refused Inputs and Forbidden Actions. Proceedings of the Workshop on Model Based Testing (MBT 2004), Elsevier, 2006.
- [17]. И.Б.Бурдонов, А.С.Косачев, В.В.Кулямин. Формализация тестового эксперимента. Программирование, 2007 г., №5, стр. 3-32.
- [18]. И.Б.Бурдонов, А.С.Косачев, В.В.Кулямин. Безопасность, верификация и теория конформности. Материалы второй международной научной конференции по проблемам безопасности и противодействия терроризму. МГУ 2006, М., МЦНМО, 2007 г., стр. 135-158.
- [19]. И.Б.Бурдонов, Косачев А.С. Системы с приоритетами: конформность, тестирование, композиция. Программирование, 2009 г., № 4, стр.24-40.
- [20]. И.Б. Бурдонов, А.С. Косачев. Тестирование с преобразованием семантик. Труды ИСП РАН, том 17, 2009 г., стр.193-208.
- [21]. A.Kossachev, I.Burdonov. Formal Conformance Verification, Short Papers of the 22nd IFIP ICTSS, Alexandre Petrenko, Adenilso Simao, Jose Carlos Maldonado (eds.), Nov. 08-10, 2010, Natal, Brazil, pp.1-6.
- [22]. А.С. Косачев, И.Б.Бурдонов. Семантики взаимодействия с отказами, дивергенцией и разрушением. Программирование, 2010 г., №5, стр. 3-23.
- [23]. И.Б.Бурдонов, А.С.Косачев. Согласование конформности и композиции. Программирование, 2013 г., №6, стр. 3-15.

Automata system: determinism conditions and testing

Igor Burdonov <igor@ispras.ru>

Alexander Kossatchev kos@ispras.ru

*Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

Abstract. The problem of testing of aggregate systems is considered. The system components are described with finite automata with multiple entries and exits. The communication between automata is described with message passing over simplex

communication channels. The system is described with an oriented graph of links. Each node of the graph corresponds to automaton of a component and an arc corresponds to a communication channel connecting exit of one automaton with entry of another automaton. The hypothesis of the links is assumed: the graph of links is static and the link structure is error-free. Automaton of the graph node in each state can accept multiple messages from its entries (at most one message from each entry) and send multiple messages to its exits (at most one message to each exit). An associative composition of the automata is defined. The restrictions on the automata and the graph of links are determined, for which their composition, i.e. the system itself, is deterministic. The goal of testing is to cover transitions of the automata reachable during the system work. It is assumed that during testing it is possible to observe the states of automata and the messages on the arcs. The complete testing of the automata system without considering the hypothesis of links may require the number of testing actions of order of multiplication of numbers of states of the component automata, while with the consideration of the hypothesis of links – of order of the sum of these numbers. If the numbers of states of all automata are equal, it gives exponential reduction of the number of test actions. If the hypothesis of links is true, an algorithm of test generation for a deterministic system is proposed basing on filtration of tests generated for covering all transitions of the composition. Test is rejected if it covers only such transitions of the components that are covered by the remaining tests. In conclusion, the directions of future research are described.

Keywords: directed graphs, graph coverage, communicating automata, automata composition, distributed systems, testing, networks.

DOI: 10.15514/ISPRAS-2016-28(1)-9

For citation: Burdonov I.B., Kossatchev A.S. Automata system: determinism conditions and testing. *Trudy ISP RAN/Proc. ISP RAS*, 2016, vol. 28, issue 1, pp. 151-184 (in Russian). DOI: 10.15514/ISPRAS-2016-28(1)-9

References

- [1]. I. B. Burdonov, A. S. Kossatchev. Testirovanie sistemy avtomatov [Testing of automata system]. *Trudy ISP RAN [The proceeding of ISP RAS]*, Vol. 28(1), 2016. (in Russian). DOI: 10.15514/ISPRAS-2016-28(1)-7
- [2]. I. B. Burdonov, A. S. Kossatchev. Sistema avtomatov: kompoziciia po grafu sviazei [Automata system: composition on connection graph]. *Trudy ISP RAN [The proceeding of ISP RAS]*, Vol. 28(1), 2016 г.. (in Russian)
- [3]. Revised Working Draft on “Framework: Formal Methods in Conformance Testing”. JTC1/SC21/WG1/Project 54/1, ISO Interim Meeting, ITU-T on, Paris, 1995.
- [4]. Bourdonov I.B., Kossatchev A.S., Kuliamin V.V. Teoriya sootvetstviya dlya system s blokirovkami i razrusheniem [Conformance theory of the systems with Refused Inputs and Forbidden Actions]. Moscow, «Nauka», 2008, 412 p. (in Russian)
- [5]. Bourdonov I. Teoriya konformnosti (funkcional'noe testirovanie prorammny'kh system na osnove formal'ny'kh modelej [Conformance theory (functional testing on formal model base)]. LAP LAMBERT Academic Publishing, Saarbrucken, Germany, 2011, ISBN 978-3-8454-1747-9, 428 p. (in Russian)
- [6]. Bourdonov I.B., Kossatchev A.S. Specification Completion for IOCO. Programming and Computer Software, vol. 37(1), 2011, pp. 1-14. DOI: 10.1134/S0361768811010014

- [7]. A. S. Kamkin, M. M. Chupilko. Survey of modern technologies of simulation-based verification of hardware. *Programming and Computer Software*, vol. 37 (3), 2011, pp. 147-152. DOI: 10.1134/S0361768811030017
- [8]. I. B. Burdonov, A. S. Kossatchev, V. V. Kuli Amin. Irredundant Algorithms for Traversing Directed Graphs: The Deterministic Case. *Programming and Computer Software*, vol. 29(5), 2003, pp. 245-258. DOI: 10.1023/A:1025733107700
- [9]. A. Levitin. *Algoritmy: vvedenie v razrabotku i analiz* [Introduction to The Design and Analysis of Algorithms]. M.: «Viliams», 2006, pp. 160-163, ISBN 0-201-74395-7. (in Russian)
- [10]. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms*. 2-nd Edition. MIT Press Cambridge, MA, USA, 2001, ISBN 0-262-03293-7.
- [11]. Bourdonov I.B., Kossatchev A.S., Kuli Amin V.V. Application of Finite Automats for Program Testing. *Programming and Computer Software*, vol. 26(2), 2000, pp. 61-73. DOI: 10.1007/BF02759192
- [12]. Bourdonov I.B., Kossatchev A.S. Complete Open-State Testing of Limitedly Nondeterministic Systems. *Programming and Computer Software*, vol. 35(6), 2009, pp.301-313. DOI: 10.1134/S0361768809060012
- [13]. Bourdonov I.B., Kossatchev A.S. Semantiki vzaimodejstviya s otkazami, divergentsiej i razrusheniem. Chast' 2. Usloviya konechnogo polnogo testirovaniya. [Semantics of Interaction with Refused Inputs, Divergence and Forbidden Actions. Part 2. The condition of finite complete testing]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika*. [Tomsk State University. Journal of Control and Computer Science], 2011, №2, pp. 89-98. (in Russian)
- [14]. Bourdonov I.B., Kossatchev A.S. Testirovanie konformnosti na osnove sootvetstviya sostoyanij [Conformance testing based on a state relation]. *Trudy ISP RAN* [The proceeding of ISP RAS], vol. 18, 2010, pp. 183-320. (in Russian)
- [15]. 84. Bourdonov I.B., Kossatchev A.S.. Safe simulation testing of systems with refusals and destructions. *Automatic Control and Computer Sciences*, vol. 45(7), 2011, pp. 380-389. DOI: 10.3103/S0146411611070042
- [16]. I.B.Bourdonov, A.S.Kossatchev, V.V.Kuli Amin. Formal Conformance Testing of Systems with Refused Inputs and Forbidden Actions. *Proceedings of the Workshop on Model Based Testing (MBT 2004)*, Elsevier, 2006.
- [17]. Bourdonov I.B., Kossatchev A.S., Kuli Amin V.V. Formalization of Test Experiments. *Programming and Computer Software*, vol. 33(5), 2007, pp. 239-260. DOI: 10.1134/S0361768807050015
- [18]. Bourdonov I.B., Kossatchev A.S., Kuli Amin V.V. Bezopasnost', verifikatsiya i teoriya konformnosti [Safety, Verification and Conformance Theory]. *Materialy Vtoroj mezhdunarodnoj nauchnoj konferentsii po problemam bezopasnosti i protivodejstviya terrorizmu* [The proceeding of the Second international conference on the problems of safety and counteraction against terrorism], Moscow, MNCMO, 2007, pp. 135-158. (in Russian)
- [19]. Bourdonov I.B., Kossatchev A.S. Systems with Priorities: Conformance, Testing, and Composition. *Programming and Computer Software*, 2009, Vol/35(4), pp.198-211. DOI: 10.1134/S0361768809040045
- [20]. Bourdonov I.B., Kossatchev A.S. Testirovanie s preobrazovaniem semantic [Testing with Semantics Conversion] *Trudy ISP RAN* [The proceeding of ISP RAS], vol. 17, 2009, pp. 193-208. (in Russian)

- [21]. A.Kossachev, I.Burdonov. Formal Conformance Verification, Short Papers of the 22nd IFIP ICTSS, Alexandre Petrenko, Adenilso Simao, Jose Carlos Maldonado (eds.), Nov. 08-10, 2010, Natal, Brazil, pp.1-6.
- [22]. Bourdonov I.B., Kossatchev A.S. Interaction Semantics with Refusals, Divergence, and Destruction. *Programming and Computer Software*, vol. 36(5), 2010, pp. 247-263. *DOI*: 10.1134/S0361768810050014
- [23]. Bourdonov I.B., Kossatchev A.S. Agreement between Conformance and Composition. *Programming and Computer Software*, vol. 39(6), 2013, pp. 269–278. *DOI*: 10.1134/S0361768813060029