

Решение автоматных уравнений для временных автоматов относительно параллельной композиции

^{1,2} О.В. Кондратьева <olga.kondratyeva@telecom-sudparis.eu>

¹ Н.В. Евтушенко <yevtushenko@sibmail.com>

² А.Р. Кавалли <ana.cavalli@telecom-sudparis.eu>

¹ НИ ТГУ, 634050, Россия, г. Томск, пр. Ленина, дом 36

² Телеком Южный Париж, 91000, Франция, г. Эври, ул. Чарльза Фурье, дом 9

Аннотация. Рассматривается задача синтеза неизвестной компоненты, которая при совместной работе с известной частью системы удовлетворяет заданной спецификации, в случае, когда поведение компонент и спецификация системы в явном виде зависят от времени. Результаты, известные для решения параллельного автоматного уравнения для классических конечных автоматов, обобщаются на случай автоматов с таймаутами. Задача синтеза неизвестной компоненты возникает во многих приложениях и в статье иллюстрируется ее применение для реализации композиции сервисов с заданными ограничениями на временные параметры качества.

Ключевые слова: временной автомат, параллельная композиция, автоматное уравнение, наибольшее решение, ограничения внешней среды.

1. Введение

При синтезе и анализе многокомпонентных дискретных систем нередко возникает задача синтеза неизвестной компоненты, которая при совместной работе с известной частью системы удовлетворяет заданной спецификации. Как показано в [1, 2] данная задача часто может быть формализована в виде решения соответствующего языкового или автоматного уравнения, однако в разработанных алгоритмах явно не учитываются временные аспекты поведения систем. В данной работе мы рассматриваем модель автомата с таймаутами, далее временной автомат [3, 4], расширенную на случай возможного недетерминизма по времени задержек обработки входных символов. В [5] уже было показано, что такое расширение оказывается необходимым при описании совместного поведения компонент, допускающих недетерминизм по входным символам.

Реальные системы всегда функционируют в некоторой среде, которая может воздействовать на систему и в которой возможно наблюдение за реакциями системы. Свойства среды влияют как на управляемость и наблюдаемость

системы, так и могут накладывать дополнительные ограничения на поведение системы. Использование таких ограничений внешней среды, как частичная управляемость, инерционность (конечность обработки сигналов средой и системой), конечность времени ожидания ответа от используемого приложения, может оказаться полезным для повышения эффективности разработки функциональных и качественных требований, для оптимизации многокомпонентных систем, а также при построении тестов. Поэтому мы рассматриваем построение композиции и решение уравнений для временных автоматов с учетом дополнительных ограничений, накладываемых на среду и/или компоненты системы.

Статья организована следующим образом: в разделе 2 вводятся необходимые определения и обозначения для автомата с таймаутами и рассматриваются ограничения для внешней среды через поведение максимального автомата. Раздел 3 посвящен построению параллельной композиции, для которой в разделе 4 приводится формула для нахождения наибольшего решения при различных ограничениях для внешней среды. В разделе 5 обсуждаются перспективные направления для дальнейших исследований.

2. Временной автомат с таймаутами

Функционирование автомата всегда осуществляется в некоторой *внешней среде*, предполагаемой или заданной явно. Свойства среды влияют как на возможность или невозможность подачи и наблюдения определенных символов (управляемость и наблюдаемость системы), так и могут накладывать дополнительные ограничения на поведение системы.

В общем случае, без дополнительных ограничений, внешняя среда, в которой функционирует временной автомат, удовлетворяет следующим предположениям и требованиям: 1) со стороны внешней среды на автомат могут поступить только символы из некоторого входного алфавита I , и со стороны автомата во внешнюю среду поступают символы из некоторого выходного алфавита O ; 2) выходной символ может быть получен внешней средой только после подачи на автомат некоторого входного символа, т.е. входной символ всегда предшествует выходному символу; 3) после получения автоматом входного символа, возможно, по истечении некоторого времени, будет произведен некоторый выходной символ; 4) внешняя среда является достаточно «медленной», т.е. следующий входной символ может быть подан на систему, поведение которой описано временным автоматом, не ранее, чем был получен выходной символ на предыдущее входное воздействие. Описанная подобным образом среда может рассматриваться как наиболее общая для функционирования автоматов с заданными входным и выходным алфавитами, так как выполнение данных требований не накладывает никаких дополнительных ограничений на поведение и структуру автоматов.

Как обычно, \mathbf{N} далее обозначает множество натуральных чисел.

Автоматом с таймаутами [3, 5] или в данной статье просто *временным автоматом* называется семёрка $S = (S, I, O, s_0, \lambda_S, \Delta_S, \sigma_S)$, где пятёрка $(S, I, O, s_0, \lambda_S)$ есть классический конечный автомат, дополненный *функцией задержки входного символа* $\Delta_S: S \rightarrow S \times (\mathbf{N} \cup \{\infty\})$ и *функцией задержки выходного символа* $\sigma_S: \lambda_S \rightarrow 2^{(\mathbf{N} \cup \{0\}) \setminus \emptyset}$, где S – конечное непустое множество состояний с выделенным начальным состоянием s_0 , I и O – конечные входной и выходной алфавиты соответственно, $\lambda_S \subseteq S \times I \times S \times O$ – отношение переходов. Функция Δ_S (далее также называемая функцией таймаута) определяет для каждого состояния максимальное время ожидания входного символа (таймаут) и состояние, в которое автомат перейдёт по истечении таймаута в случае, если ни один входной символ не поступил. Функция σ_S определяет для каждого перехода непустое, возможно бесконечное множество допустимых интервалов времени, необходимых для выполнения соответствующего перехода и получения выходного символа в ответ на поступившее входное воздействие (выходной таймаут). Аналогично [5], далее мы предполагаем, что множество значений σ_S для каждого перехода может быть представлено множеством значений конечного набора целочисленных линейных функций вида $\{b + k \cdot t \mid b, k \in \{0\} \cup \mathbf{N}\}$.

Свойства детерминированности и полной определенности формулируются для временных автоматов аналогично конечным автоматам. Автомат называется *полностью определенным*, если для любого $s_1 \in S$ и любого $i \in I$ существует, по крайней мере, одна пара $(s_2, o) \in S \times O$, такая, что $(s_1, i, s_2, o) \in \lambda_S$, в противном случае автомат называется *частичным*. Временной автомат называется *i-детерминированным*, или *детерминированным по входным символам*, если для любых состояния $s_1 \in S$ и входного символа $i \in I$ определено не более одной пары $(s_2, o) \in S \times O$ такой, что $(s_1, i, s_2, o) \in \lambda_S$, в противном случае, автомат называется *i-недетерминированным*. Временной автомат называется *t-детерминированным*, или *детерминированным по времени*, если для каждого перехода $(s_1, i, s_2, o) \in \lambda_S$ определено единственное значение допустимой задержки выхода, т.е. $|\sigma_S((s_1, i, s_2, o))| = 1$, в противном случае, автомат называется *t-недетерминированным*. Временной автомат называется *детерминированным*, если он *i*- и *t*-детерминированный, в противном случае – *недетерминированным*.

Поведение временного автомата существенно зависит от временных параметров, поэтому для его описания вводятся понятия временных входного и выходного символов [3]. *Временной входной символ* $(i, t) \in I \times (\{0\} \cup \mathbf{N})$, поданный на автомат в текущем состоянии s , означает, что входной символ i подается на автомат по истечении t единиц времени с момента достижения автоматом состояния s или по истечении t единиц времени с момента наблюдения предыдущего выходного символа. *Временной выходной символ* $(o, k) \in O \times (\{0\} \cup \mathbf{N})$ показывает, что символ o выдается автоматом через k единиц времени после получения входного символа. Для вычисления реакции автомата на временной входной символ (i, t) , поданный в состоянии s , на

основе таймаутов вычисляется состояние, в которое автомат перейдет из состояния s по истечении времени t [3]: входной символ i поступит на автомат в состоянии s_p , в котором оканчивается последовательность переходов по таймаутам $\Delta_S(s) = (s_1, T_1)$, $\Delta_S(s_1) = (s_2, T_2)$, ..., $\Delta_S(s_{p-1}) = (s_p, T_p)$ такая, что $T_1 + T_2 + \dots + T_{p-1} < t$, но $T_1 + T_2 + \dots + T_p \geq t$. Отметим, что если $\Delta_S(s) = (s, \infty)$, то $s_p = s$ при любом t . Таким образом, в результате подачи символа (i, t) в состоянии s автомат может выдать выходной символ o через k единиц времени, если и только если существует переход $(s_p, i, s', o) \in \lambda_S$ и $k \in \sigma_S((s_p, i, s', o))$, и в этом случае временной выходной символ (o, k) есть реакция автомата на временной входной символ (i, t) в состоянии s .

Рассмотрим временную входную последовательность $\alpha = (i_1, t_1)(i_2, t_2) \dots (i_n, t_n)$ и временную выходную последовательность $\beta = (o_1, k_1)(o_2, k_2) \dots (o_n, k_n)$. Если в автомате S существует такая последовательность переходов, что выходная последовательность β может быть реакцией автомата в состоянии s на входную последовательности α , то пара $\alpha\beta$ называется *временной вход-выходной последовательностью* или *временной трассой* автомата S в состоянии s . Множество всех временных трасс автомата S в состоянии s обозначим $trace_S(s)$.

Два состояния s и p временных автоматов S и P называются *эквивалентными*, если множества трасс в этих состояниях совпадают, то есть $trace_S(s) = trace_P(p)$. Автоматы S и P называются *эквивалентными*, если их начальные состояния эквивалентны. Автомат S есть *редукция* автомата P , обозначение $S \leq P$, если $trace_S(s_0) \subseteq trace_P(p_0)$.

Множество всех временных трасс автомата S может быть представлено соответствующим конечным полуавтоматом $Aut(S)$ с выделенным специальным символом $1 \notin I \cup O$ для обозначения ожидания в течение единицы времени, такого, что полуавтомат $Aut(S)$ распознает последовательность $\alpha = 1^{t_1} i_1 1^{k_1} o_1 \dots 1^{t_m} i_m 1^{k_m} o_m$, если и только если $(i_1, t_1).(o_1, k_1) \dots (i_m, t_m).(o_m, k_m)$ есть временная трасса автомата S [5]. Конструктивно, соответствующий полуавтомат может быть построен заменой каждого перехода временного автомата на цепочку переходов, в которой переходы по входному и выходному символу разделены требуемым числом переходов по символу 1. Переходы по таймауту представляются цепочкой переходов по символу 1 через «копии» начального состояния цепочки.

Утверждение 1 [5]. Временные автоматы S и P эквивалентны, если и только если эквивалентны соответствующие им полуавтоматы $Aut(S)$ и $Aut(P)$. Автомат S есть редукция автомата P , если и только если $Aut(S)$ есть редукция $Aut(P)$.

Язык, распознаваемый полуавтоматом $Aut(S)$, будем называть (*развернутым по времени*) *языком временного автомата* S и обозначать $L(S)$.

Максимальным временным автоматом в алфавитах I и O будем называть такой временной автомат $M_{I,O}$, который содержит в себе любой временной автомат S как редукцию. Язык максимального временного автомата обозначим $L(M_{I,O}) = [(1)^* I (1)^* O]^* (1)^*$. По определению, язык $L(S)$ некоторого временного автомата является $[(1)^* \cup (1)^* I (1)^* O]$ -префикс-замкнутым подмножеством языка $L(M_{I,O})$, то есть таким, что для любых последовательностей $a \in L(M_{I,O})$ и $b \in [(1)^* \cup (1)^* I (1)^* O]$ справедливо: если $ab \in L(S)$, то и $a \in L(S)$. Отметим, что язык полуавтомата, соответствующего временному автомату, распознает не только трассы временного автомата, но и все их продолжения с суффиксами из $(1)^*$. Данные суффиксы отражают тот факт, что после выдачи последнего выходного символа автомат не прекращает свое функционирование и не осуществляет сброс в начальное состояние (если не сказано иное), и может продолжать изменять состояние под действием таймаутов.

Максимальным временным автоматом в заданной среде будем называть максимальный подавтомат $M_{I,O}$, удовлетворяющий ограничениям, связанным в частности с дополнительными ограничениями внешней среды, в которой функционирует временной автомат.

Полезными на практике оказываются следующие ограничения и их комбинации:

1. *Инерционность внешней среды*: после получения от автомата выходного символа, среде требуется некоторое минимальное время на генерацию следующего входного символа. Язык максимального автомата в инерционной среде будет иметь вид $[(1)^* I (1)^* O (1)^n]^* (1)^*$, где $n \in \mathbf{N}$ представляет минимальное время реакции среды.
2. *Инерционность системы*: любой системе требуется некоторое минимальное отличное от нуля время, чтобы произвести выходную реакцию на полученное входное воздействие. Язык максимального автомата для инерционных систем будет иметь вид $[(1)^* I (1)^m (1)^* O]^* (1)^*$, где $m \in \mathbf{N}$ представляет минимальное время реакции системы.
3. *Ограниченность времени ожидания*: среда может ожидать выходную реакцию системы на поданное входное воздействие не дольше некоторого заранее заданного интервала времени. Язык

максимального автомата для систем в такой среде будет иметь вид $[(1)^* I(1)^{\leq k} O]^*(1)^*$, где $k \in \mathbf{N}$ представляет максимальное время ожидания реакции системы. Использование данного предположения на практике требует, во-первых, явного описания поведения внешней среды в случаях нарушения времени ожидания (например, перезагрузка системы, прерывание текущей операции, и т.п.), во-вторых, интерпретации отсутствия выходного символа по истечении заданного времени ожидания.

4. *Запрет на подачу входной последовательности* $\alpha = (i_1, t_1) \dots (i_n, t_n)$: на временной автомат не может быть подана входная последовательность α в силу свойств внешней среды или если данная последовательность приводит к недопустимому поведению системы. Данное требование полностью описывается рассмотрением редукций соответствующего частичного максимального автомата, имеющего язык вида $[(1)^* I(1)^* O]^*(1)^* / [(1)^{t_1} i_1 (1)^* O] \dots [(1)^{t_n} i_n (1)^* O] (1)^*$.

Можно рассмотреть более сложные ограничения для внешней среды, однако далее мы предполагаем, что для любого рассматриваемого ограничения на внешнюю среду, если временной автомат M удовлетворяет предъявленному ограничению, то и любая редукция автомата M также ему удовлетворяет. В таком случае ограничения среды могут быть представлены в виде соответствующего максимального временного автомата, то есть автомата, описывающего максимально возможное поведение при соблюдении допустимых ограничений.

3. Параллельная композиция временных автоматов

Для описания взаимодействия временных автоматов в режиме диалога вводится операция параллельной композиции. Схема бинарной параллельной композиции приведена на рис. 1. Без ограничения общности, далее мы полагаем, что алфавиты I_1, I_2, O_1, O_2, U, V попарно не пересекаются. Обозначим через $I = I_1 \cup I_2$ и $O = O_1 \cup O_2$ внешние входной и выходной алфавиты композиции.

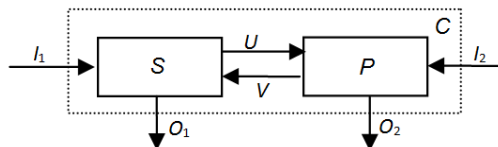


Рис. 1. Схема параллельной композиции

Аналогично композиции для конечных автоматов [1], параллельная композиция временных автоматов может быть построена по соответствующим полуавтоматам. Параллельной композицией [1, 2] полуавтоматов $A = (A, X \cup Y, a_0, \delta_A, F_A)$ и $B = (B, Y \cup Z, b_0, \delta_B, F_B)$ называется полуавтомат $C = (C, X \cup Z, c_0, \delta_C, F_C)$, такой, что $C = [A \upharpoonright_{Z/X} \cap B \upharpoonright_{X/Z}] \downharpoonright_{X \cup Z}$ (обозначение: $C = A \diamond B$), где операции пересечения (\cap), расширения (\upharpoonright) и ограничения (\downharpoonright) полуавтоматов определены известным образом на основе соответствующих операций над регулярными языками [1, 2].

Пусть заданы временные автоматы $S = (S, I_1 \cup V, O_1 \cup U, s_0, \lambda_S, \Delta_S, \sigma_S)$ и $P = (P, I_2 \cup U, O_2 \cup V, p_0, \lambda_P, \Delta_P, \sigma_P)$ и ограничения внешней среды для композиции выражены автоматом $M_C \leq M_{I,O}$. Тогда временной автомат C , которому соответствует полуавтомат $Aut(C) = [Aut(S) \diamond Aut(P)] \cap Aut(M_C)$, есть параллельная композиция временных автоматов S и P при ограничении M_C . В том случае, если дополнительные ограничения для внешней среды не заданы, то есть $M_C = M_{I,O}$, то определение параллельной композиции совпадает с таковым из [5].

Утверждение 2. Если $S_1 \leq S_2$ и $P_1 \leq P_2$, то $S_1 \diamond P_1 \leq S_2 \diamond P_2$.

Доказательство следует из свойств операций (расширение, ограничение, пересечение) над языками, используемых для определения операции композиции.

Следствие. Максимальный автомат композиции любых компонент S и P , удовлетворяющих ограничениям, выраженным максимальными автоматами M_S и M_P соответственно, есть автомат $M_C = M_S \diamond M_P$. В частности, композиция инерционных автоматов (с ненулевой минимальной задержкой выхода) также будет инерционным автоматом.

4. Наибольшее решение уравнения для временных автоматов

Рассмотрим автоматное уравнение $A \diamond X \sim C$, коэффициенты A и C которого описываются временными автоматами, \diamond есть оператор параллельной композиции, а в качестве отношения конформности \sim выступает редукция (в этом случае можно говорить об автоматном неравенстве). Тогда по определению параллельной композиции, неравенство для временных автоматов вида

$$A \diamond X \leq C$$

можно решать как неравенство над соответствующими полуавтоматами

$$Aut(A) \diamond Aut(X) \cap Aut(M_C) \leq Aut(C),$$

которое сводится к следующему неравенству над языками полуавтоматов

$$L(A) \diamond L(X) \cap L(M_C) \subseteq L(C).$$

Далее, применяя известное преобразование $A \cup B = C \Leftrightarrow A = C \cap \bar{B}$ из теории множеств, получим языковое неравенство вида

$$L(A) \diamond L(X) \subseteq L(C) \cup \overline{L(M_C)}.$$

Используя известные результаты для наибольшего решения языкового неравенства [1], найдем наибольшее решение: $L_{Solution} = \overline{L(A) \diamond (\overline{L(C)} \cap L(M_C))}$. В общем случае, язык $L_{Solution}$ не является языком некоторого временного автомата, но являясь наибольшим решением, содержит в себе в том числе и все решения, соответствующие временным автоматам и являющиеся решениями исходного автоматного уравнения. Наибольшее решение $L_{Solution-TFSM}$, являющееся языком временного автомата и, следовательно, решением автоматного уравнения, есть наибольшее префикс-замкнутое подмножество языка $L_{Solution} \cap L(M_X)$, где M_X есть максимальный автомат для неизвестной компоненты. Для краткости, обозначим I_X и O_X входной и выходной алфавиты неизвестной компоненты, соответственно.

Утверждение 3. Наибольшее решение автоматного уравнения (неравенства) $A \diamond X \leq C$ для временных автоматов есть временной автомат S такой, что $L(S)$ есть наибольшее $[(1)^* \cup (1)^* I_X (1)^* O_X]$ -префикс-замкнутое подмножество языка $L_{Solution} \cap L(M_X)$, где $L_{Solution} = \overline{L(A) \diamond (\overline{L(C)} \cap L(M_C))}$ и M_X есть максимальный автомат для неизвестной компоненты. Если $L_{Solution} = \emptyset$, то уравнение неразрешимо. Если $L_{Solution} \neq \emptyset$, но $L(S) = \emptyset$, то уравнение не имеет автоматного решения.

Доказательство. Пересечение с языком $L(M_X)$ обеспечивает, что решение, если существует, содержится в максимально возможном допустимом поведении автомата для неизвестной компоненты. Алгоритм построения наибольшего префикс-замкнутого подмножества заданного регулярного языка известен [1]. Следовательно, $L(S)$ есть максимальный язык временного автомата, содержащийся в $L_{Solution}$ и являющийся решением автоматного уравнения (неравенства).

Временной автомат, соответствующий языку $L(S)$, в общем случае представляет собой недетерминированный, возможно частичный, автомат, описывающий совокупность всех возможных допустимых реализаций неизвестной компоненты. Стоит заметить, однако, что частичность решения

требует особого внимания: каждая реализация на практике должна быть полностью определенной, поэтому особый интерес представляет вопрос решения уравнений и неравенств для полностью определенных автоматов, а также вопросы интерпретации и адекватного до определения частичных и недетерминированных решений.

Заметим, что в силу того, что языковые уравнения решаются одинаковым образом для любых коэффициентов-языков, описанный процесс построения наибольшего решения уравнения для временных автоматов является универсальным при любых ограничениях внешней среды для композиции, если данные ограничения могут быть выражены через поведение максимального временного автомата в заданной среде.

5. Пример

В качестве примера для иллюстрации решения уравнений для временных автоматов рассмотрим задачу реализации композиции двух сервисов. При разработке сервиса в виде композиции других сервисов возникают следующие задачи: 1) проверить, предоставляет ли взаимодействие заданных компонент требуемый сервис, и 2) осуществить выбор компонент, которые бы во взаимодействии предоставляли желаемый сервис. В том случае, если проектируемый сервис и сервисы-компоненты описаны временными автоматами с таймаутами, данные задачи могут быть решены, в том числе, и с учетом требований к временным параметрам качества композиции сервисов.

Рассмотрим в качестве примера простую композицию для заказа товара в пункт самовывоза: покупатель осуществляет заказ (request) и по истечении оговоренного времени доставки может забрать заказ из пункта самовывоза (deliver). Время доставки для такого сервиса является одним из наиболее важных параметров качества сервиса. Потребуем, чтобы время доставки не превышало 7 дней. Тогда временной автомат, приведенный на рис. 2 (слева), описывает заданную композицию с указанным требованием к качеству сервиса (ограничением максимального времени доставки).



Рис. 2. Временной автомат для сервиса заказа с самовывозом с ограничением времени доставки (слева) и без ограничения времени доставки (справа)

Заданная композиция может быть реализована композицией двух сервисов: сервиса *Продавец* и сервиса *Пункт самовывоза*. Опишем взаимодействие между сервисами: покупатель заказывает некоторый товар у *Продавца* (request) и *Продавец* отправляет запрос на проверку наличия товара в *Пункте самовывоза* (check). Если товар есть в наличии (product), то покупателю отправляется уведомление о возможности получить заказ (deliver), иначе отправляется запрос на поставку товара со склада (ship). После поставки

товара в *Пункт самовывоза* (get_product), после проверки (check) и подтверждения, что был поставлен необходимый товар (product), покупатель получает уведомление о доставке, либо в случае ошибки (no_product) *Продавец* повторно заказывает доставку товара. На рис. 3 приведен временной автомат, описывающий работу *Продавца*.

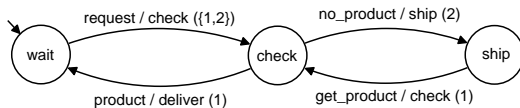


Рис. 3. Временной автомат для компоненты *Продавец*

Найдем решение уравнения для неизвестной компоненты *Пункт самовывоза* без ограничения времени доставки, но при условии, что никакое действие не может быть осуществлено моментально и минимальная задержка выходного сигнала составляет 1 день (предположение об инерционности системы). Наибольшее решение представлено на рис. 4. Решение существует, следовательно, описанная композиция может быть реализована. На рис. 4 символ * использован для обозначения любого произвольного поведения (т.е. множества всех возможных входо-выходных пар автомата с любой возможной задержкой выхода), и соответствует части поведения компоненты, которая не может быть достигнута из заданного контекста; соответственно, эта часть не влияет на результат композиции и потому может быть определена (реализована) любым оптимальным образом. Символ \mathbb{N} обозначает множество натуральных чисел.

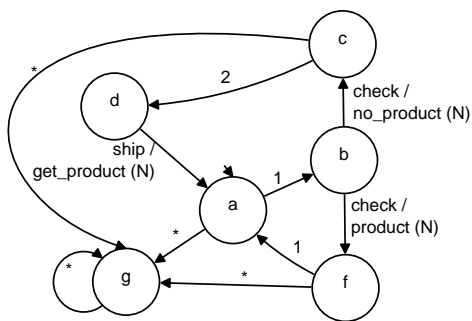


Рис. 4. Наибольшее решение для компоненты *Пункт самовывоза*

Оптимизируем работу *Пункта самовывоза* таким образом, чтобы минимизировать время доставки товара, и проверим, при каких условиях возможно гарантировать выполнение требования о доставке в течение 7 дней. Решение (полностью определенное) с минимальным временем задержки приведено на рис. 5 (слева) и композиция данного частного решения с контекстом приведена на рис. 5 (справа). Из построенной композиции видно, что при заданном контексте (*Продавец*) выбранное решение для компоненты

Пункт самовывоза не может гарантировать соблюдение ограничения по максимальному времени доставки в силу возможности возникновения осцилляции (о чем свидетельствует наличие периода $5t_b$ множестве задержек выхода композиции без ограничения верхнего значения). Дальнейший более детальный анализ композиции может показать, что композиция с ограничением времени доставки в 7 дней будет гарантировано выполняться только в том случае, если запрашиваемый продукт всегда будет находиться в *Пункте самовывоза* (т.е. соответствующая компонента в ответ на запрос *check* всегда будет отвечать *product*). В иных ситуациях требуется либо пересмотр заданных требований к качеству, либо дальнейшая оптимизация компонент композиции, включая дальнейшую оптимизацию работы компоненты *Продавец*.

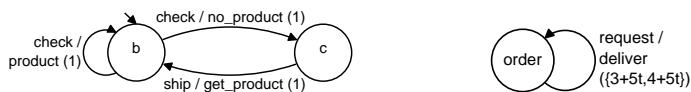


Рис. 5. Оптимальное по времени поведение компоненты *Пункт самовывоза* (слева) и ее композиция с компонентой *Продавец* (справа)

6. Заключение

В статье рассмотрено построение параллельной композиции временных автоматов и синтеза неизвестной компоненты, которая при взаимодействии с известной частью системы удовлетворяет заданной спецификации композиции. На примере простой композиции для заказа товара с доставкой в пункт самовывоза проиллюстрированы перспективы использования введенных операций для оценки и управления качеством композиции сервисов.

Отметим, что значения функций таймаутов и задержек выхода в рассматриваемой модели временного автомата могут принимать только целочисленные значения. При рассмотрении реальных систем (в частности, сервисов) подобное ограничение оказывается достаточно жестким: для задержек выхода, более реалистичной интерпретацией является определение интервала времени, в течение которого автомат может выдать выходной символ. Так как в композиции задержки выходного символа одной компоненты определяют моменты времени поступления внутренних символов на вход другой компоненты, то расширение области значений функции задержки выходного символа на произвольные интервалы, в общем случае, требует дальнейших исследований. Однако все результаты, полученные при целочисленных задержках выходного символа, сохраняются при использовании открытых справа интервалов, то есть при замене каждой целочисленной задержки n на интервал $[n, n + 1)$. Это достигается за счет того, что при целочисленных таймаутах любой входной символ, поступивший на автомат в некоторый момент времени в интервале $[n, n + 1)$, обрабатывается автоматом так же, как входной символ, поступивший точно в момент времени

л. Подобные рассуждения согласуются с выводами работы [4] при сравнении выразительности моделей временного автомата с таймаутами и интервалами. Использование параллельной композиции и решения соответствующих уравнений для временных автоматов представляется перспективным для решения таких задач, как тестирование в контексте, и анализ свойств композиции взаимозаменяемых компонент, например, при оценке качества динамической композиции сервисов. Также интересной для дальнейшего исследования проблемой является вопрос разрешимости автоматного уравнения при различных ограничениях внешней среды.

Список литературы

- [1]. T. Villa, N. Yevtushenko, R.K. Brayton, A. Mishchenko, A. Petrenko, A.L. Sangiovanni Vincentelli. The unknown component problem: theory and applications. Springer, 2012. 311 p.
- [2]. Н.В. Евтушенко, А.Ф. Петренко, М.В. Ветрова. Недетерминированные автоматы: анализ и синтез. Ч.1. Отношения и операции: Учебное пособие. Томск: Томский государственный университет, 2006. 142 с.
- [3]. M. Zhigulin, N. Yevtushenko, S. Maag, A.R. Cavalli. FSM-based test derivation strategies for systems with timeouts. Proceedings of the international conference QSIC 2011. P. 141-149.
- [4]. D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power. Proceedings GandALF 2014. EPTCS 161, 2014. P. 203-216.
- [5]. О.В. Кондратьева, Н.В. Евтушенко, А.Р. Кавалли. Параллельная композиция конечных автоматов с таймаутами. Вестн. Том. гос. ун-та. Управление, вычислительная техника и информатика. № 2(27). 2014. стр. 73–81.

Solving parallel equations for Finite State Machines with Timeouts

^{1,2} O. Kondratyeva <olga.kondratyeva@telecom-sudparis.eu>

¹ N. Yevtushenko <yevtushenko@sibmail.com>

² A. Cavalli <ana.cavalli@telecom-sudparis.eu>

¹ Tomsk State University, 634050, Russia, Tomsk, av. Lenina, 36

² Telecom SudParis, 91000, France, Evry, rue Charles Fourier 9

Abstract. The problem of designing an unknown component that combined with the known part of a discrete event system satisfies the given overall specification arises in a number of applications. In this paper, we extend the known results for classical Finite State Machines (FSM) to Finite State Machines with Timeouts (TFSM).

A TFSM is an FSM augmented with an input and an output timeout functions, prescribing the change of current state if no input is applied until a specified timeout expires and system delays needed to response to applied input, respectively. We represent the behavior of a TFSM by the corresponding regular language, so that $L(S)$ has a word $\alpha = 1t1\ i1\ 1k1\ o1\ \dots\ 1tm\ im\ 1km\ om$ if and only if the given TFSM S has a timed trace $(i1, t1).(o1, k1)\ \dots\ (im, tm).(om, km)$. The parallel composition of two TFSMs $C = S \diamond P$ is defined via composition of languages $L(C) = [L(S) \diamond L(P)] \cap L(MC)$, where MC is a maximal TFSM over input and output alphabets of the composition.

The unknown component X is then designed as a solution to the equation $A \diamond X \leq C$ with TFSM coefficients A (context) and C (specification), where \leq is the reduction relation which specifies that the behavior of a system behavior to be designed is contained in that of the specification. Similar to classical FSMs, the equation is transformed to a language equation

$L(A) \diamond L(X) \subseteq L(C) \cup \overline{L(M_C)}$ with the largest solution

$L_{Solution} = L(A) \diamond \left(\overline{L(C)} \cap L(M_C) \right)$, the largest TFSM subset of which provides the largest solution to the solvable TFSM equation. After the largest solution is derived, a corresponding reduction can be extracted in order to provide the component with required properties.

The application areas vary from testing in context to quality optimization of service compositions. Future work includes studying equation solvability criteria and properties of nondeterministic and partial specifications and solutions.

Keywords: finite state machine with timeouts, parallel composition, automata equations, the largest solution, environment restrictions.

References

- [1]. T. Villa, N. Yevtushenko, R.K. Brayton, A. Mishchenko, A. Petrenko, A.L. Sangiovanni Vincentelli. The unknown component problem: theory and applications. Springer, 2012. 311 p.
- [2]. N.V. Yevtushenko, A.F. Petrenko, M.V. Vetrova. Nedeterminirovannye avtomaty: analiz i sintez. Ch.1. Otnosheniya i operazii [Nondeterministic finite state machines: analysis and synthesis. Part 1. Relations and operations]: Tutorial, Tomsk: Tomsk State University, 2006. 142 p. (in Russian).

- [3]. M. Zhigulin, N. Yevtushenko, S. Maag, A.R. Cavalli. FSM-based test derivation strategies for systems with timeouts. Proceedings of the international conference QSIC 2011. P. 141-149.
- [4]. D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power. Proceedings GandALF 2014. EPTCS 161, 2014. P. 203-216.
- [5]. O.V. Kondratyeva, N.V. Yevtushenko, A.R. Cavalli. Parallel composition of nondeterministic Finite State Machines with Timeouts // Tomsk State University Journal of Control and Computer Science. 2014. № 2(27). P. 73–81. (in Russian).