

DOI: 10.15514/ISPRAS-2025-37(6)-1



О длине адаптивной различающей последовательности для семейства наблюдаемых автоматов

¹ И.Б. Бурдонов, ORCID: 0000-0001-9539-7853 <igor@ispras.ru>

^{1,2} Н.В. Евтушенко, ORCID: 0000-0002-4006-1161 <evtushenko@ispras.ru>

¹ А.С. Косачев, ORCID: 0000-0001-5316-3813 <kos@ispras.ru>

¹ Институт системного программирования РАН им. В.П. Иванникова,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

² Национальный исследовательский университет «Высшая школа экономики»,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. В статье исследуется длина адаптивной различающей последовательности для семейства инициальных полностью определенных наблюдаемых, возможно, недетерминированных автоматов. Приводится верхняя оценка длины такой последовательности и показывается ее достижимость. Обсуждается, каким образом полученные результаты можно применить при построении адаптивных диагностических тестов на основе конечно-автоматной модели.

Ключевые слова: недетерминированные автоматы; адаптивная различающая последовательность; диагностические тесты.

Для цитирования: Бурдонов И.Б., Евтушенко Н.В., Косачев А.С. О длине адаптивной различающей последовательности для семейства наблюдаемых автоматов. Труды ИСП РАН, том 37, вып. 6, часть 1, 2025 г., стр. 7–20. DOI: 10.15514/ISPRAS-2025-37(6)-1.

On Length of an Adaptive Distinguishing Sequence for a Family of Observable Finite State Machines

¹ I.B. Burdonov, ORCID: 0000-0001-9539-7853 <igor@ispras.ru>

^{1,2} N.V. Yevtushenko, ORCID: 0000-0002-4006-1161 <evtushenko@ispras.ru>

¹ A.S. Kossatchev, ORCID: 0000-0001-5316-3813 <kos@ispras.ru>

¹ Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

² National Research University Higher School of Economics,
20, Myasnitskaya st., Moscow, 101000, Russia.

Abstract. In the paper, the length of an adaptive distinguishing sequence for a family of initialized complete observable, possibly nondeterministic Finite State Machines (FSM) is studied. The upper bound for the length of such a sequence is established and its reachability is shown. It is also discussed how the obtained results can be applied for constructing adaptive diagnostic test suites based on a FSM model.

Keywords: nondeterministic FSMs; adaptive distinguishing sequence; diagnostic test suites.

For citation: Burdonov I.B., Yevtushenko N.V., Kossatchev A.S. On length of an adaptive distinguishing sequence for a family of observable Finite State Machines. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 6, part 1 2025, pp. 7-20 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-1.

1. Введение

В настоящей статье рассматривается задача синтеза диагностических тестов на основе модели инициального, возможно, недетерминированного конечного автомата. Предполагается, что после выполнения этапа тестирования с использованием проверяющих тестов известно конечное множество автоматов, которые соответствуют «неправильной» реализации некоторых переходов в автомате-спецификации. Для определения, какая именно реализация была предъявлена в процессе тестирования, то есть какие исправления необходимо внести в предъявленную реализацию, чтобы реализация соответствовала известной спецификации, используются диагностические тесты. Под *диагностическим* тестом понимается входная последовательность (или конечное множество входных последовательностей), по реакции на которые можно различить любые две реализации из множества конечных автоматов, зафиксированных после тестирования на соответствие спецификации, и соответственно распознать тестируемый автомат-реализацию. В работе обсуждается построение адаптивной входной последовательности, по реакции на которую можно различить любую пару из множества рассматриваемых автоматов-реализаций. В первом алгоритме для построения адаптивной различающей последовательности строится прямая сумма всех автоматов множества и исследуется возможность построения адаптивной различающей последовательности для множества начальных состояний прямой суммы. В работе оценивается наибольшая длина адаптивной различающей последовательности, если такая последовательность существует, и показывается, что если число состояний каждого из k автоматов-реализаций равно n , и адаптивная различающая последовательность для прямой суммы автоматов-реализаций существует, то длина такой последовательности может достигать величины n^k . Полученная оценка существенно меньше оценки длины различающей последовательности для произвольного автомата с nk состояниями, k из которых являются начальными [1]. Во втором алгоритме для каждой пары «реализация_1-реализация_2» строится адаптивная различающая последовательность, по поведению на которую различаются эти автоматы-реализации, и из таких различающих последовательностей «собирается» адаптивная различающая последовательность для множества всех автоматов-реализаций. Оценивается длина такой последовательности, которая не превышает величины $n^2k + k$, если число состояний каждого из k автоматов-реализаций равно n .

Структура статьи следующая. Второй раздел содержит необходимые определения и обозначения. Третий раздел описывает связь между диагностическими тестами и различающими последовательностями. В четвертом разделе устанавливаются условия существования адаптивной различающей последовательности для множества начальных состояний прямой суммы автоматов-реализаций, и оценивается длина такой последовательности (если существует). В пятом разделе предлагается алгоритм построения адаптивной различающей последовательности для множества начальных состояний автоматов-реализаций на основе «сборки» такой последовательности для пар различных реализаций. В заключении освещаются перспективы дальнейших научных исследований. В частности, отмечается, что предлагаемый подход можно использовать для синтеза диагностических тестов для определенного класса входов-выходных полуавтоматов.

2. Основные определения и обозначения

Под *конечным автоматом* (или просто *автоматом*) понимается пятерка $S = (S, I, O, h_S, S_{in})$ [2-3], где S – конечное непустое множество состояний с выделенным непустым множеством начальных состояний S_{in} , I и O – конечные непустые входной и выходной алфавиты соответственно, такие, что $I \cap O = \emptyset$, и $h_S \subseteq S \times I \times O \times S$ – отношение или множество переходов. Автомат называется *инициальным*, если S_{in} – одноэлементное множество, то есть $|S_{in}| = 1$. Автомат называется *неинициальным*, если $S_{in} = S$, и тогда в ряде случаев автомат обозначается $S = (S, I, O, h_S)$. В случае, когда $1 < |S_{in}| < |S|$, автомат называется *слабо инициальным*. Четверка $(s, i, o, s') \in h_S$ называется *переходом* в автомате в состоянии s или *переходом* из состояния s в состояние s' . Говорят, что *поведение автомата* S *определено* в состоянии s для входного символа i , если существует пара $(o, s') \in O \times S$ такая, что $(s, i, o, s') \in h_S$. Если поведение в автомате S определено в любом состоянии $s \in S$ для любого входного символа $i \in I$, то автомат называется *полностью определенным*, в противном случае автомат называется *частично определенным* или *частичным*. Автомат S называется *детерминированным*, если для любой пары $(s, i) \in S \times I$ существует не более одной пары $(o, s') \in O \times S$ такой, что $(s, i, o, s') \in h_S$. В противном случае автомат называется *недетерминированным*. Если в недетерминированном автомате S для любой тройки $(s, i, o) \in S \times I \times O$ существует не более одного состояния $s' \in S$ такого, что $(s, i, o, s') \in h_S$, то автомат называется *наблюдаемым*, в противном случае автомат называется *ненаблюдаемым*. В настоящей работе мы рассматриваем полностью определенные наблюдаемые автоматы, если явно не сказано иное.

Множество состояний $\{s' \in S \mid (s, i, o, s') \in h_S\}$ называется *io-преемником* состояния $s \in S$ для $i \in I, o \in O$. Вообще говоря, *io-преемник* состояния s может быть пустым множеством; в этом случае иногда говорят, что *io-преемник* состояния s *не существует*. В наблюдаемом автомате для каждого состояния $s \in S$ и каждой входов-выходной пары i/o соответствующий *io-преемник* содержит не более одного состояния. Для наблюдаемого полностью определенного автомата можно ввести функцию следующего состояния и функцию выходов: функция $next_states(s, i)$ определяет возможное множество состояний автомата после подачи входного символа i в состоянии s , функция $next_state_s^{nd}(s, i, o)$ определяет состояние автомата после подачи входного символа i в состоянии s при получении выходного символа o , и функция выходов $outs(s, i)$ определяет возможное множество выходных символов автомата после подачи входного символа i в состоянии s .

Отношение переходов обычным образом распространяется на последовательности (слова) в алфавитах I и O . Обозначим через I^* множество всех последовательностей конечной длины в алфавите I , включая пустую последовательность ε . Соответственно, функции переходов $next_states$, $next_state_s^{nd}$ и выходов $outs$ можно распространить на последовательности входных и выходных символов. В этом случае множество $next_states(s, \alpha)$, $\alpha \in I^*$, включает

те и только те состояния, которые достижимы в автомате S из состояния s по входной последовательности α , то есть $next_states(s, \alpha)$ есть α -преемник состояния s . Соответственно, множество $outs(s, \alpha)$ включает все возможные выходные последовательности (реакции) автомата S в состоянии s на входную последовательность α . Пара $\alpha/\beta, \alpha \in I^*, \beta \in outs(s, \alpha)$, называется *входо-выходной последовательностью* автомата S в состоянии s . По определению, выходной реакцией автомата на пустую входную последовательность в любом состоянии является пустая выходная последовательность. Функция $next_state_s^{nd}(s, \alpha, \beta), \alpha = i_1 \dots i_k$ и $\beta = o_1 \dots o_k$, определяет $\alpha\beta$ -преемник, то есть состояние, которое достижимо в автомате S из состояния s по входной последовательности α с выходной последовательностью β . Если $\beta \notin outs(s, \alpha)$, то $next_state_s^{nd}(s, \alpha, \beta) = \emptyset$ или в этом случае говорят, что $\alpha\beta$ -преемник состояния s не существует.

Состояния s_1 и s_2 полностью определенного автомата $S = (S, I, O, h_S, S_{in})$ называются *неразделимыми* или в данной работе *неразличимыми* (обозначение: $s_1 \sim s_2$) [4-5], если $\forall \alpha \in I^* (outs(s_1, \alpha) \cap outs(s_2, \alpha) \neq \emptyset)$; в противном случае, состояния s_1 и s_2 называются *различимыми* (обозначение: $s_1 \not\sim s_2$). Последовательность α , для которой $outs(s_1, \alpha) \cap outs(s_2, \alpha) = \emptyset$, называется *разделяющей* или *различающей* последовательностью для этих состояний. Если $|S_{in}| > 1$, то входная последовательность, которая различает каждую пару начальных состояний автомата, называется *различающей* последовательностью автомата. Автоматы $S = (S, I, O, h_S, S_{in})$ и $P = (P, I, O, h_P, P_{in})$ называются *неразличимыми* (обозначение: $S \sim P$), если для любой входной последовательности α существуют состояния $s \in S_{in}$ и $p \in P_{in}$ такие, что $outs(s, \alpha) \cap outs(p, \alpha) \neq \emptyset$; в противном случае, автоматы S и P называются *различимыми* (обозначение: $S \not\sim P$). Входная последовательность α , для которой $outs(s, \alpha) \cap outs(p, \alpha) = \emptyset$ для любых $s \in S_{in}$ и $p \in P_{in}$, называется *различающей* последовательностью для этих автоматов. Заметим, что введенное определение различающей последовательности для детерминированных автоматов совпадает с хорошо известным [7-8]; для недетерминированных автоматов введенное выше понятие различимости не требует использования «всех погодных условий», то есть при анализе различимости двух недетерминированных автоматов достаточно подать различающую последовательность только один раз.

Прямой суммой полностью определенных инициальных автоматов S и $P, S = (S, I, O, h_S, s_0), P = (P, I, O, h_P, p_0), S \cap P = \emptyset$, называется автомат $S \oplus P = (S \cup P, I, O, h_S \cup h_P, \{s_0, p_0\})$, и это определение естественным образом расширяется на прямую сумму k автоматов, $k > 2$. Прямая сумма инициальных автоматов $A_1 \oplus \dots \oplus A_k$ с попарно непересекающимися множествами состояний $S_j = \{a_j^1, \dots, a_j^{n_j}\}, j = 1, \dots, k, k > 1$, и начальными состояниями a_1^1, \dots, a_k^1 , получается «склеиванием» автоматов A_j , имеет множество состояний, которое является объединением множеств $S_j, j = 1, \dots, k$, множество начальных состояний $\{a_1^1, \dots, a_k^1\}$, входной алфавит I и выходной алфавит O .

Как было отмечено во введении, при подаче входной последовательности, различающей два (или несколько) автоматов-реализаций, можно распознать, на какой из этих автоматов была подана эта последовательность без использования предположения о «всех погодных условиях», то есть с использованием таких различающих последовательностей можно говорить о диагностических тестах, если при тестировании на основе «белого ящика» обнаружено несоответствие тестируемой реализации спецификации. Поскольку известно, что безусловные различающие последовательности для недетерминированных автоматов могут оказаться достаточно длинными [5], то при синтезе диагностических тестов для таких автоматов имеет смысл вместо безусловных различающих последовательностей рассматривать адаптивные различающие последовательности, которые существуют чаще и обычно оказываются более короткими. Для представления адаптивных входных последовательностей используются тестовые примеры [1, 6].

Тестовый пример определяется для заданных входного и выходного алфавитов I и O . Далее инициальный автомат называется *связным*, если любое состояние автомата достижимо из начального состояния по некоторой входной последовательности. Состояние автомата называется *нетупиковым*, если в состоянии есть переход по некоторому входному символу; иначе, состояние называется *тупиковым*. Тестовый пример $TC(I, O)$ есть связный наблюдаемый инициальный автомат над алфавитами I и O , граф переходов которого ациклический, и в каждом нетупиковом состоянии определены переходы только по одному входному символу со всеми возможными выходными символами (в англоязычной литературе – *single-input output-complete*). По определению, если $|I| > 1$, то тестовый пример является частичным автоматом. Тестовый пример представляет адаптивную входную последовательность, и состояния s_1 и s_2 полностью определенного наблюдаемого автомата $S = (S, I, O, h_S)$ называются *адаптивно различимыми* (далее просто *различимыми*), если существует тестовый пример $TC(I, O)$, такой что каждая входо-выходная последовательность, ведущая в тупиковое состояние этого автомата, реализуется в только в одном из состояний s_1 и s_2 , то есть соответствующая адаптивная входная последовательность различает состояния s_1 и s_2 . Инициальные полностью определенные наблюдаемые автоматы (S, I, O, h_S, s_0) и (T, I, O, h_T, t_0) называются *адаптивно различимыми* (далее просто *различимыми*), если существует тестовый пример $TC(I, O)$, такой что каждая входо-выходная последовательность, ведущая в тупиковое состояние этого автомата, не реализуется в обоих состояниях s_0 и t_0 , то есть соответствующая адаптивная входная последовательность различает автоматы S и T . Если тестовый пример является различающим для автоматов S и T , то в дальнейшем тупиковые состояния тестового примера будут помечаться символами \bar{S} , \bar{T} , (\bar{S}, \bar{T}) . Пометка тупикового состояния символом \bar{S} означает, что соответствующая входо-выходная последовательность автомата $TC(I, O)$ не реализуется в начальном состоянии автомата S , то есть такую последовательность нельзя наблюдать при подаче на автомат S адаптивной входной последовательности, задаваемой этим тестовым примером. Аналогично, пометка тупикового состояния символом \bar{T} означает, что соответствующая входо-выходная последовательность автомата $TC(I, O)$ не реализуется в начальном состоянии автомата T , то есть такую последовательность нельзя наблюдать при подаче на автомат T адаптивной входной последовательности, задаваемой тестовым примером $TC(I, O)$. Согласно пометке тупикового состояния символом (\bar{S}, \bar{T}) , соответствующая входо-выходная последовательность автомата $TC(I, O)$ не реализуется ни в начальном состоянии автомата S , ни в начальном состоянии автомата T , то есть такую последовательность нельзя наблюдать при подаче на автомат S или автомат T адаптивной входной последовательности, задаваемой тестовым примером $TC(I, O)$.

Тестовый пример $TC(I, O)$ называется *различающим* для слабо-инициального автомата $S = (S, I, O, h_S, S_{in})$, если каждая входо-выходная последовательность, ведущая в тупиковое состояние автомата $TC(I, O)$, реализуется не более, чем в одном состоянии множества S_{in} , то есть соответствующая адаптивная входная последовательность различает каждую пару начальных состояний автомата S . Для автомата S существует адаптивная различающая последовательность, если и только если для автомата S существует различающий тестовый пример [6]. Под высотой тестового примера или длиной соответствующей адаптивной входной последовательности понимается длина самой длинной последовательности из начального в тупиковое состояние, то есть длина самой длинной входной последовательности, которая будет подана на автомат в процессе различающего эксперимента. Различающие последовательности автомата-спецификации активно используются при оптимизации тестов с гарантированной полнотой на основе автоматной модели [см., например, 9 и 10], и как известно [11-12], адаптивные различающие последовательности короче безусловных и существуют чаще.

Один из подходов к синтезу различающих тестовых примеров для наблюдаемых недетерминированных автоматов базируется на понятии *адаптивной различимости* или

просто *различимости* для подмножеств состояний автомата S , которое определяется итеративно [1]. Все синглеты множества S полагаются *0-различимыми*. Пусть $l \geq 1$ и определены все максимальные (в смысле вложенности) $(l-1)$ -различимые подмножества множества S . Подмножество S' состояний автомата *l-различно*, если S' есть $(l-1)$ -различимое множество или существует входной символ $i \in I$, такой, что для любого $o \in O$, io -преемник множества S' пуст или является $(l-1)$ -различимым множеством, причем в последнем случае любые два различных состояния из S' не могут обладать одним и тем же io -преемником. Входной символ $i \in I$, такой, что для любого $o \in O$, io -преемник множества S' пуст или является $(l-1)$ -различимым множеством, причем любые два различных состояния из S' не обладают одним и тем же io -преемником, называется *различающим входным символом* для S' . Множество S' называется *адаптивно различимым* или в данной работе просто *различимым*, если S' есть l -различимое множество для некоторого натурального l . По определению различающего входного символа имеет место следующее утверждение.

Утверждение 1. Если для множества $S' \subseteq S$ существуют входной символ i и выходной символ o , такие что io -преемник множества S' есть S' , то i не является различающим входным символом для S' .

Известно, что слабо инициальный полностью определенный наблюдаемый автомат S обладает различающим тестовым примером [1], если и только если множество его начальных состояний адаптивно различимо.

3. Диагностические тесты и различающие последовательности

При синтезе проверяющих и диагностических тестов на основе модели конечного автомата предполагается, что есть, возможно, недетерминированный автомат-спецификация S и некоторое множество его реализаций. Если в процессе тестирования выясняется, что предъявленный автомат-реализация не соответствует автомату-спецификации, причиной чего достаточно часто является «неправильная» реализация некоторых переходов в автомате-спецификации, то возникает задача обнаружения таких переходов с целью «исправления» предъявленного автомата-реализации. В этом случае необходимо решить задачу построения входной (адаптивной) последовательности, различающей каждую пару автоматов-реализаций, если такая последовательность существует, то есть построить *диагностический тест*. Предполагая, что при реализации автомата-спецификации «подозрительных» переходов, которые могут быть неправильно реализованы, не слишком много, для построения такой входной последовательности решается следующая задача. Пусть есть k реализаций автомата-спецификации, каждая из которых имеет n состояний. Если любая пара «реализация_1-реализация_2» (адаптивно) различима, то подав соответствующие входные последовательности на тестируемый автомат-реализацию, можно определить, что нужно «поправить» в соответствующем автомате-реализации. Если не все пары реализаций адаптивно различимы, то вывод можно сделать с некоторой вероятностью.

4. Условия существования адаптивной различающей последовательности для семейства инициальных автоматов-реализаций и оценка ее длины, если такая последовательность существует

В настоящем разделе нас интересует вопрос о существовании входной последовательности, различающей любую пару автоматов из известного семейства автоматов-реализаций. Мы далее полагаем, что семейство автоматов-реализаций, предъявленное для построения диагностического теста, содержит k инициальных полностью определенных наблюдаемых автоматов, каждый из которых, так же как спецификация, имеет n состояний. Автомат-реализация есть автомат A_j с множеством состояний $S_j = \{a_j^1, \dots, a_j^n\}$, $n \geq 2$, $j = 1, \dots, k$, с

начальным состоянием a_j^1 . Входной и выходной алфавиты каждого автомата A_j , $j = 1, \dots, k$, совпадают с таковыми алфавитами I и O автомата-спецификации. Без ограничения общности будем полагать, что множества состояний двух автоматов A_j , $j = 1, \dots, k$, для различных значений j не пересекаются. Для распознавания предъявленной реализации с целью исправления в ней ошибок мы используем (адаптивную) различающую последовательность, то есть входную последовательность α , которая различает любую пару автоматов из множества A_j , $j = 1, \dots, k$, (если такая последовательность существует). После подачи адаптивной различающей последовательности на предъявленную реализацию, по реакции на эту последовательность можно обнаружить, какой именно автомат-реализация A_j , $j = 1, \dots, k$, требует корректировки. По определению различающей последовательности, справедливо следующее утверждение.

Утверждение 2. Входная последовательность α , которая различает любую пару автоматов из множества A_j , $j = 1, \dots, k$, существует, если и только если существует различающая последовательность для прямой суммы автоматов A_j , $j = 1, \dots, k$.

К сожалению, известно [5], что длина безусловной входной последовательности, различающей инициальные автоматы с n состояниями, может достигать 2^{n^2-1} , и соответственно длина различающей последовательности для прямой суммы $k \geq 2$ инициальных автоматов с n состояниями может также достигать 2^{n^2-1} , в то время как длина адаптивной различающей последовательности для двух инициальных автоматов с n состояниями равна n^2 [6, 12].

Рассмотрим прямую сумму $A_1 \oplus \dots \oplus A_k$ из k инициальных автоматов, $k \geq 2$, в которой каждый автомат A_j , $j = 1, \dots, k$, имеет множество состояний $S_j = \{a_j^1, \dots, a_j^n\}$ с начальным состоянием a_j^1 . Множество состояний $\{a_{j_1}^{r_1}, \dots, a_{j_l}^{r_l}\} \in S_{j_1} \times \dots \times S_{j_l}$, $1 < l \leq k$, где $j_1, \dots, j_l \in \{1, \dots, k\}$, попарно различны, является l -различимым, если существует входной символ i , на который выходные реакции прямой суммы $A_{j_1} \oplus \dots \oplus A_{j_l}$ различаются в любых двух состояниях множества $\{a_{j_1}^{r_1}, \dots, a_{j_l}^{r_l}\}$. Пусть $l > 1$ и определены все $(l-1)$ -различимые подмножества множества S . Подмножество $S' \in S_{j_1} \times \dots \times S_{j_l}$, $1 < l \leq k$, где $j_1, \dots, j_l \in \{1, \dots, k\}$ попарно различны, l -различимо, если S' есть $(l-1)$ -различимое множество или для S' существует различающий входной символ $i \in I$, такой, что для любого $o \in O$, io -преобразованное множества S' является синглетоном или $(l-1)$ -различимым множеством. Если множество $\{a_1^1, \dots, a_k^1\}$ есть l -различимое множество для некоторого натурального l , то начальные состояния a_1^1, \dots, a_k^1 прямой суммы $A_1 \oplus \dots \oplus A_k$ различимы посредством адаптивного эксперимента высоты l . Таким образом, длина адаптивной различающей последовательности для множества начальных состояний такой прямой суммы не превосходит самой длинной последовательности по io -преобразованным из множества начальных состояний. Поскольку из множества начальных состояний прямой суммы достижимы только элементы декартова произведения множеств состояний автоматов-компонентов, то имеет место следующее утверждение.

Утверждение 3. Длина адаптивной различающей последовательности для прямой суммы k инициальных полностью определенных наблюдаемых автоматов с n состояниями, $n \geq 2$, $k \geq 2$, не превосходит величины n^k .

Заметим, что оценка в утверждении 3 существенно меньше оценки достижимой длины $(2^{nk} - nk - 1)$ адаптивной различающей последовательности для произвольного автомата с nk состояниями, k из которых являются начальными [1]. Например, при $k = 3$, $n = 2$, имеет место $2^{nk} - nk - 1 = 57$, в то время как сумма чисел из утверждения 3 равна 2^3 .

Доказательство достижимости оценки из утверждения 3 похоже на доказательство достижимости оценки для произвольного случая, и далее мы показываем, что для любого $k \geq 2$ существуют k инициальных автоматов, каждый из которых имеет n состояний, $n \geq 2$, такие,

что для прямой суммы этих автоматов кратчайший различающий тестовый пример имеет высоту n^k .¹

Модифицируя результаты работы [1], на множестве всех подмножеств из k состояний, все состояния которых принадлежат различным автоматам A_j , вводится отношение линейного порядка, представленное цепью $C(n, k)$. Цепь $C(n, k)$ строится итеративно, начиная с $k = 1$, и по определению, $C(n, 1) = \{a_1^1\}, \{a_1^2\}, \dots, \{a_1^n\}$. Если уже построена цепь $C(n, k)$, то для построения цепи $C(n, k+1)$ в каждое подмножество цепи $C(n, k)$ добавляется состояние a_{k+1}^1 , после чего получается цепь $C^1(n, k+1)$. Если уже построена цепь $C^j(n, k+1)$, $1 < j < n$, то для получения цепи $C^{j+1}(n, k+1)$ подмножества в цепи $C^j(n, k+1)$ записываются в обратном порядке и в каждое подмножество вместо состояния a_{k+1}^j добавляется состояние a_{k+1}^{j+1} . Цепь $C(n, k+1)$ получается как конкатенация цепей $C^j(n, k+1)$, $j = 1, \dots, k+1$.

В качестве примера рассмотрим автоматы A_1, A_2, A_3 с множествами состояний $S_1 = \{1, 2\}$, $S_2 = \{a, b\}$, $S_3 = \{A, B\}$, т.е. $n = 2$, $k = 3$. По описанным выше правилам $C(2, 1) = \{1\}, \{2\}$. Для построения цепи $C^1(2, 2)$ в каждое множество цепи $C(2, 1)$ добавляем состояние a : $C^1(2, 2) = \{1, a\}, \{2, a\}$. Чтобы построить цепь $C^2(2, 2)$ меняем порядок в цепи $C^1(2, 2)$ и вместо состояния a записываем состояние b : $C^2(2, 2) = \{2, b\}, \{1, b\}$. После конкатенации цепей $C^1(2, 2)$, $C^2(2, 2)$ получаем цепь $C(2, 2) = \{1, a\}, \{2, a\}, \{2, b\}, \{1, b\}$, в которой любые два соседних блока отличаются только одним состоянием. Аналогично, при построении $C(2, 3)$ строим цепи $C^1(2, 3)$, $C^2(2, 3)$, $C^3(2, 3)$ и их конкатенацию: $C^1(2, 3) = \{1, a, A\}, \{2, a, A\}, \{2, b, A\}, \{1, b, A\}$, $C^2(2, 3) = \{1, b, B\}, \{2, b, B\}, \{2, a, B\}, \{1, a, B\}$, и после конкатенации $C^1(2, 3)$ и $C^2(2, 3)$ получаем цепь $C(2, 3) = \{1, a, A\}, \{2, a, A\}, \{2, b, A\}, \{1, b, A\}, \{1, b, B\}, \{2, b, B\}, \{2, a, B\}, \{1, a, B\}$, в которой есть любая тройка состояний автоматов A_1, A_2, A_3 , и любые соседние блока отличаются только одним состоянием.

Утверждение 4. Цепь $C(n, k)$, $1 \leq n$, $1 \leq k$, содержит каждое подмножество из k состояний, состоящее из состояний автоматов-компонентов, причем любые соседние блоки отличаются только одним состоянием.

Доказательство того, что в цепи $C(n, k)$ есть все подмножества из k состояний, содержащие состояния всех автоматов-компонентов, и любые два соседних блока отличаются только одним состоянием, достаточно просто проводится с использованием индукции по числу k автоматов-компонентов.

Введенное цепью $C(n, k)$ отношение линейного порядка на множестве подмножеств состояний автоматов-компонентов обозначим символом $\succ_{C(n,k)}$. Если подмножество p_{j+1} непосредственно следует за p_j в цепи $C(n, k)$, то $p_j \succ_{C(n,k)} p_{j+1}$. По цепи $C(n, k)$, мощность которой равна n^k , построим автомат $\mathcal{S}_{C(n,k)}$, который есть прямая сумма автоматов A_j , $j = 1, \dots, k$, каждый из которых имеет n состояний. Число входных и выходных символов $\mathcal{S}_{C(n,k)}$ совпадает с числом элементов в цепи $C(n, k) = p_1, \dots, p_{C(n,k)}$. Для последнего подмножества используется специальный входной символ i_{dist} , и, кроме того, используются $(n^k - 1)$ входных символов, каждый из которых имеет вид $i_{p/p'}$ для соседних блоков p и p' цепи $C(n, k)$ (табл. 1 и 2). Множество выходных символов содержит выходные символы $\langle 0 \rangle, \langle 1 \rangle, \dots, \langle k \rangle$, а также все выходные символы вида $\langle p \rangle, p \in C(n, k)$. Для каждой пары p и p' множеств в цепи $C(n, k)$, такой, что множество $p' = p \setminus \{j\} \cup \{r\}$ следует за p , то есть $p \succ_{C(n,k)} p'$, введем входной символ $i_{p/p'}$, по которому из состояния j есть переход в состояние r с выходным символом $\langle 0 \rangle$. Кроме того, в каждом состоянии s множества $p \setminus \{j\}$ есть петля с выходным символом $\langle 0 \rangle$. Таким образом, $i_{p/p'} \langle 0 \rangle$ -преемником множества p является множество p' . В каждом состоянии

¹ Подобное утверждение можно сформулировать и для случая, когда автоматы прямой суммы имеют различное число состояний n_1, \dots, n_k . В настоящей статье мы предполагаем, что в качестве элементов прямой суммы используются достаточно простые автоматы-мутанты, и число состояний каждого автомата-мутанта совпадает с числом состояний n автомата-спецификации.

$s \notin p$ по входному символу $i_{p/p'}$ есть петля с выходным символом $\langle p'' \rangle$ для каждого $p'' \in C(n, k) \setminus p$, $s \in p$. Для состояния $s \in p \setminus \{j\}$ есть петля с выходным символом $\langle p'' \rangle$ для каждого $p'' \in C(n, k)$, $s \in p''$ и $p'' \neq p$.

Табл. 1. Фрагмент таблицы переходов автомата $\mathcal{S}_{C(2, 3)}$ по входному символу $i_{1aA/2aA}$.

Table 1. A frame of the transition table of $\mathcal{S}_{C(2, 3)}$ under input $i_{1aA/2aA}$.

Состояние	1	2	a	b	A	B
Входной символ	2/(0);	$\langle 2aA \rangle$;	a/(0);	$\langle 1bA \rangle$;	B/(0);	$\langle 1aB \rangle$;
$i_{1aA/2aA}$	$\langle 1aB \rangle$;	$\langle 2aB \rangle$;	$\langle 1aB \rangle$;	$\langle 1bB \rangle$;	$\langle 2aA \rangle$;	$\langle 1bB \rangle$;
	$\langle 1bA \rangle$;	$\langle 2bA \rangle$;	$\langle 2aA \rangle$;	$\langle 2bA \rangle$;	$\langle 2bA \rangle$;	$\langle 2aB \rangle$;
	$\langle 1bB \rangle$	$\langle 2bB \rangle$	$\langle 2aB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$

Табл. 2. Фрагмент таблицы переходов автомата $\mathcal{S}_{C(2, 3)}$ по входному символу i_{dist} .

Table 2. A frame of the transition table of $\mathcal{S}_{C(2, 3)}$ under input i_{dist} .

Состояние	1	2	a	b	A	B
Входной символ	1/(1);	2/ $\langle 2aB \rangle$;	a/(a);	b/ $\langle 1bA \rangle$;	A/ $\langle 1aA \rangle$;	B/(B);
i_{dist}	$\langle 1aA \rangle$;	$\langle 2aA \rangle$;	$\langle 1aB \rangle$;	$\langle 1bB \rangle$;	$\langle 2aB \rangle$;	$\langle 2aB \rangle$;
	$\langle 1bA \rangle$;	$\langle 2bA \rangle$;	$\langle 1aA \rangle$;	$\langle 2bA \rangle$;	$\langle 2bA \rangle$;	$\langle 1bB \rangle$;
	$\langle 1bB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$	$\langle 2bB \rangle$

Соответственно, $i_{p/p'}$ -преемником множества p'' цепи $C(n, k)$, отличного от p , является множество p'' . Поэтому если множество p' является t -разделимым, то множество p является $(t+1)$ -разделимым, поскольку по входному символу $i_{p/p'}$ из p достижимо p' с выходным символом (0), а по любому другому входному символу из p достижимо p с выходным символом $\langle p \rangle$. Чтобы разделить состояния последнего множества $p_{C(n, k)}$ цепи $C(n, k)$ добавляется входной символ i_{dist} , который разделяет состояния последнего подмножества выходными символами (1), ..., (k). Для состояния $s \notin p_{C(n, k)}$ есть петля с выходным символом $\langle p'' \rangle$ для каждого $p'' \in C(n, k)$, $s \in p''$. Для состояния $s \in p_{C(n, k)}$ есть петля с выходным символом $\langle p'' \rangle$ для каждого $p'' \in C(n, k)$, $s \in p''$ и $p'' \neq p_{C(n, k)}$. Обозначим построенный таким образом автомат $\mathcal{S}_{C(n, k)}$. По построению, для автомата $\mathcal{S}_{C(n, k)}$ высота кратчайшего тестового примера совпадает с длиной цепи $C(n, k)$, то есть справедливо следующее утверждение.

Утверждение 5. Множество начальных состояний автомата $\mathcal{S}_{C(n, k)}$, который имеет kn состояний, n^k входных символов и $(n^k + k + 1)$ выходных символов², является n^k -различимым. Этот факт достаточно просто доказывается, поскольку входной символ $i_{p/p'}$ будет различающим входным символом только для подмножества p . Для всех остальных подмножеств $p'' \neq p$ входной символ $i_{p/p'}$ не является различающим, поскольку по входному символу $i_{p/p'}$ достижимо подмножество p'' за счет петель по выходному символу $\langle p'' \rangle$ (утверждение 1). Для последнего множества p_{last} цепи $C(n, k)$ используется входной символ i_{dist} , который среди множеств цепи $C(n, k)$ различает только состояния множества p_{last} , то есть это единственное множество цепи, которое является 1-различимым.

Утверждение 6. Начальные состояния автомата $\mathcal{S}_{C(n, k)}$ адаптивно различимы только тестовым примером длины n^k .

В качестве примера рассмотрим автоматы A_1, A_2, A_3 с множествами состояний $S_1 = \{1, 2\}$, $S_2 = \{a, b\}$, $S_3 = \{A, B\}$, прямая сумма которых имеет множество состояний $\{1, 2, a, b, A, B\}$. Рассмотрим цепь $C(2, 3) = \{1, a, A\}, \{2, a, A\}, \{2, b, A\}, \{1, b, A\}, \{1, b, B\}, \{2, b, B\}, \{2, a, B\}, \{1, a, B\}$ и переход в автомате $\mathcal{S}_{C(2, 3)}$ под действием входного символа $i_{1aA/2aA}$. По определению, для этого входного символа будут использованы выходные символы (0), и $\langle p'' \rangle$, $p'' \in C(2, 3)$, $p'' \neq p$.

² Подобно [1], число выходных символов можно сделать полиномиальным относительно числа состояний, но в этом случае доказательство утверждения 5 будет менее «прозрачным».

Непосредственной проверкой можно убедиться, что $i_{1aA/2aA}(0)$ -преемником множества $\{1, a, A\}$ является множество $\{2, a, A\}$. Любое другого подмножество N является $i_{1aA/2aA}<N>$ -преемником самого себя. Например, множество $\{2, a, A\}$ есть $i_{1aA/2aA}<2aA>$ -преемник самого себя. Для входного символа i_{dist} фрагмент таблицы переходов автомата $S_{C(2, 3)}$ имеет вид таблицы 2. Множество $\{2, a, B\}$ является 1-различимым входным символом i_{dist} ; любое другого подмножество N является $i_{1dist}<N>$ -преемником самого себя.

Согласно утверждению 6, при построении прямой суммы всех k автоматов A_j , высота различающего тестового примера, то есть длина адаптивной различающей последовательности, для множества начальных состояний прямой суммы может достигать n^k , то есть экспоненциальная оценка длины адаптивной различающей последовательности для множества начальных состояний автоматов-компонентов относительно числа состояний автоматов-компонентов является достижимой. Поэтому ниже мы предлагаем рассмотреть несколько другой подход к построению адаптивной различающей последовательности. Подход основан на «надстраивании» определенным образом различающих последовательностей для пар автоматов-мутантов, для которых высота кратчайшей адаптивной различающей последовательности не превосходит n^2 .

5. Построение адаптивной различающей последовательности для семейства инициальных автоматов-реализаций на основе различающих последовательностей для пар автоматов-реализаций

Если автоматы-компоненты A_j , $j = 1, \dots, k$, попарно различимы, то для каждой пары инициальных автоматов A_1, \dots, A_k , $k \geq 2$, существует различающий тестовый пример. Для пары A_j, A_r , $j, r = 1, \dots, k$, $j \neq r$, рассмотрим кратчайший различающий тестовый пример $TC(A_j, A_r)$, тупиковые вершины которого помечены \bar{A}_j, \bar{A}_r , (\bar{A}_j, \bar{A}_r) . Пометка тупикового состояния \bar{A}_j , (\bar{A}_j, \bar{A}_r) в тестовом примере $TC(A_j, A_r)$ означает, что наблюдаемая входо-выходная последовательность из начального состояния в данное тупиковое состояние не принадлежит автомату A_r (автомату A_j , автоматам (A_j, A_r)). Поскольку каждый из тестовых примеров является ациклическим автоматом, все тестовые примеры можно собрать в единое тестовое множество (или тест), добавляя в каждом тупиковом состоянии множества, построенного для уже рассмотренных тестовых примеров, сначала символ «сброс», а затем начальное состояние следующего тестового примера. В этом случае окончательный тестовый пример будет «собран» из $k(k - 1)/2$ тестовых примеров для пар автоматов-реализаций, высота каждого из которых не превосходит n^2 . В данном разделе мы рассматриваем этот подход, когда на автомат-реализацию адаптивно подаются тестовые примеры $TC(A_j, A_r)$, разделенные сигналом «сброс».

Таким образом, предлагается рассматривать множество $TC = \{TC(A_j, A_r): j, r = 1, \dots, k, j \neq r\}$. Множество $TC = \{TC(A_j, A_r): j \neq r\}$ содержит тестовый пример для любой пары A_j, A_r , $j \neq r$, поэтому после подачи всех тестовых примеров из множества автоматов A_j , $j = 1, \dots, k$, останется только один автомат A_m , который не появился с пометкой \bar{A}_m после подачи любого тестового примера. Этот автомат A_m и описывает предъявленную реализацию.

Adaptive_identification { /* Адаптивная идентификация предъявленного автомата-реализации A из семейства из множества $\{A_1, \dots, A_k\}$, $k \geq 2$, инициальных полностью определенных наблюдаемых автоматов посредством подачи входных последовательностей

*/

| Для каждой пары автоматов $A_j, A_r \in \{A_1, \dots, A_k\}$, $j, r = 1, \dots, k$, $j \neq r$,
 | Построить кратчайший различающий тестовый пример $TC(A_j, A_r)$, тупиковые вершины которого помечены \bar{A}_j, \bar{A}_r , (\bar{A}_j, \bar{A}_r) ;

```

Если для некоторой пары автоматов различающего тестового примера не существует
|   выдать сообщение «предъявленный автомат  $A$  нельзя идентифицировать
однозначно»;
|   конец алгоритма;
Иначе
|    $M := \{A_1, \dots, A_k\}$ ;
|   Пока  $|M| > 1$ 
|   |   Выбрать пару автоматов  $A_j, A_r \in M, j \neq r$ ;
|   |   Подать тестовый пример  $TC(A_j, A_r)$  на тестируемый автомат  $A$ ;
|   |   Если в  $TC(A_j, A_r)$  достижимо состояние, помеченное  $\bar{A}_j$ ,
|   |   |    $M := M \setminus A_j$ ;
|   |   Если в  $TC(A_j, A_r)$  достижимо состояние, помеченное  $\bar{A}_r$ ,
|   |   |    $M := M \setminus A_r$ ;
|   |   Если в  $TC(A_j, A_r)$  достижимо состояние, помеченное  $(\bar{A}_j, \bar{A}_r)$ ,
|   |   |    $M := M \setminus \{A_j, A_r\}$ ;
|   |   Конец цикла «Пока»;
|   Если  $M = \{A_m\}$ 
|   |    $A$  эквивалентен  $A_m$  /* для идентификации предъявлен автомат  $A_m$  */
|   Конец алгоритма; }

```

Пример. Пусть $M = \{A_1, A_2, A_3, A_4\}$, для каждой пары различных автоматов из M существует различающий тестовый пример, и для распознавания предъявлен автомат A_2 .

Если на распознаваемый автомат подан тестовый пример $TC(A_1, A_2)$, то наблюдаемая входо-выходная последовательность соответствует автомату A_2 , т.е. достигается состояние \bar{A}_1 .

Удаляем A_1 из множества M : $M = \{A_2, A_3, A_4\}$.

Пусть далее на распознаваемый автомат A_2 подан тестовый пример $TC(A_3, A_4)$.

Если наблюдаемая входо-выходная последовательность не соответствует ни автомату A_3 , ни автомату A_4 , то в тестовом примере после наблюдения входо-выходной последовательности будет достигнуто состояние (\bar{A}_3, \bar{A}_4) . Из множества M удаляются автоматы A_3 и A_4 : $M = \{A_2\}$, то есть для распознавания предъявлен автомат A_2 .

Если наблюдаемая входо-выходная последовательность соответствует автомату A_3 , то, поскольку подается тестовый пример $TC(A_3, A_4)$, входо-выходная последовательность не соответствует автомату A_4 . Поэтому в тестовом примере после наблюдения входо-выходной последовательности будет достигнуто состояние \bar{A}_4 : $M = \{A_2, A_3\}$, и т.д.

Утверждение 7. Если в результате выполнения алгоритма не выдается сообщение «предъявленный автомат A нельзя идентифицировать однозначно», то автомат A идентифицируется однозначно, и длина тестовой последовательности для идентификации автомата A не превышает $n^2k + k$.

Доказательство. Автомат A идентифицируется однозначно, поскольку наблюдаемые входо-выходные последовательности не соответствуют реализациям, удаляемым из множества M . Оценка высоты кратчайшего тестового примера, равна n^2 , количество используемых тестовых примеров не превосходит $(k - 1)$, между которыми подается сигнал «сброс».

Согласно оценке длины идентификационной последовательности (утверждение 7), второй подход к идентификации автомата, описывающего поведение предъявленной реализации, в общем случае оказывается более эффективным. Тем не менее, если существует, например, входной символ, различающий все пары начальных состояний автоматов-реализаций, то именно он и будет искомым тестовым примером. Поэтому обсуждение свойств множества автоматов-реализаций, для которых более удобным является каждый из подходов, требует дальнейших исследований. Кроме того, построение одного различающего тестового примера представляет интерес для решения других задач идентификации состояний.

6. Заключение

В настоящей статье авторы обсудили возможность построения диагностических тестов для семейства автоматов, которые являются наиболее «подозрительными» для наличия ошибок после проверки тестируемой реализации на соответствие спецификации. Показано, что длина различающей адаптивной последовательности для множества из k таких автоматов с n состояниями, $k > 1$, $n > 1$, может достигать величины n^k , как известно, для безусловных различающих последовательностей достижимая оценка еще выше. Для понижения достижимой оценки в работе предлагается еще один алгоритм построения адаптивной различающей последовательности, основанный на «надстраивании» определенным образом различающих последовательностей для пар автоматов-мутантов и повторном использовании сигнала «сброс».

Если для множества «подозрительных» автоматов-реализаций различающая последовательность не существует, то есть если в множестве существуют две адаптивно неразличимые реализации, то имеет смысл говорить об установлении допустимого множества автоматов, которому может принадлежать предъявленный автомат-реализация. Автоматы из этого множества можно различить более короткими тестами относительно эквивалентности и/или редукции, используя предположение «о всех погодных условиях». Альтернативой этому может быть идентификация предъявленного автомата-реализации с некоторой вероятностью. В последующих работах авторы предполагают исследовать оба направления.

Авторы также отмечают, что результаты работы могут быть использованы для синтеза диагностических тестов для входо-выходных полуавтоматов [13], что является еще одним направлением будущих исследований.

Список литературы / References

- [1]. Khaled El-Fakih, Nina Yevtushenko, Natalia Kushik. Adaptive distinguishing test cases of nondeterministic finite state machines: test case derivation and length estimation. *Formal Aspects Comput.*, 2018, 30(2), pp. 319-332.
- [2]. Tiziano Villa, Timothy Kam, Robert K. Brayton, and Alberto Sangiovanni-Vincentelli. *Synthesis of Finite State Machines*. Springer, 1997.
- [3]. Villa T, Yevtushenko N, Brayton R, Mishchenko A, Petrenko A, Sangiovanni-Vincentelli A. *The Unknown Component Problem: Theory and Applications*, 2012. Springer.
- [4]. P. Starke, *Abstract Automata*. American Elsevier, 1972.
- [5]. Spitsyna N. Studying the separability relation between finite state machines. *Softw. Test., Verif. Reliab.*, 2007, (17(4), pp. 227-241.
- [6]. Alexandre Petrenko, Nina Yevtushenko. Conformance Tests as Checking Experiments for Partial Nondeterministic FSM. *International Workshop on Formal Approaches to Software Testing (FATES)*, 2005, pp. 118-133.
- [7]. Moore E. F. Gedanken-experiments on sequential machines. In *Automata Studies (Annals of Mathematical Studies)*, 1956, 1, pp. 129-153.
- [8]. Гилл А. Введение в теорию конечных автоматов. Наука, 1966.
- [9]. Chow T. S. Testing software Design Modelled by Finite State Machines. *IEEE Trans. Software Eng.*, 1978, 4(3), pp. 178-187.
- [10]. D. Lee, M. Yannakakis. Testing Finite-State Machines: State Identification and Verification. *IEEE Transactions on Computers*, 1994, 43(3), pp. 306-320.
- [11]. R. Alur, C. Courcoubetis, M. Yannakakis. Distinguishing tests for nondeterministic and probabilistic machines. *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, 1995, pp. 363-372.
- [12]. H.Yenigun, N. Yevtushenko, N. Kushik, J. López. The effect of partiality and adaptivity on the complexity of FSM state identification problems. *Труды ИСП РАН*, 2018, 30(1), с. 7-24.
- [13]. N.Yevtushenko, I.Burdonov, A.Kossachev. Deriving Distinguishing Sequences for Input/Output Automata. *Proceedings of the 2020 IEEE East-West Design & Test Symposium*, 2020, pp. 1-5.

Информация об авторах / Information about authors

Игорь Борисович БУРДОНОВ – доктор физико-математических наук, главный научный сотрудник ИСП РАН. Научные интересы: формальные спецификации, генерация тестов, технология компиляции, системы реального времени, операционные системы, объектно-ориентированное программирование, сетевые протоколы, процессы разработки программного обеспечения.

Igor Borisovich BURDONOV – Dr. Sci. (Phys.-Math.), a Leading Researcher of ISP RAS. Research interests: formal specifications, test generation, compilation technology, real-time systems, operating systems, object-oriented programming, network protocols, software development processes.

Нина Владимировна ЕВТУШЕНКО, доктор технических наук, профессор, главный научный сотрудник ИСП РАН, до 1991 года работала научным сотрудником в Сибирском физико-техническом институте. С 1991 г. работала в ТГУ профессором, зав. кафедрой, зав. лабораторией по компьютерным наукам. Её исследовательские интересы включают формальные методы, теорию автоматов, распределённые системы, протоколы и тестирование программного обеспечения.

Nina Vladimirovna YEVTUSHENKO, Dr. Sci. (Tech.), Professor, a Leading Researcher of ISP RAS, worked at the Siberian Scientific Institute of Physics and Technology as a researcher up to 1991. In 1991, she joined Tomsk State University as a professor and then worked as the chair head and the head of Computer Science laboratory. Her research interests include formal methods, automata theory, distributed systems, protocol and software testing.

Александр Сергеевич КОСАЧЕВ – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН. Научные интересы: формальные спецификации, генерация тестов, технология компиляции, системы реального времени, операционные системы, объектно-ориентированное программирование, сетевые протоколы, процессы разработки программного обеспечения.

Alexander Sergeevitch KOSSATCHEV – Cand. Sci. (Phys.-Math.), a Leading Researcher of ISP RAS. Research interests: formal specifications, test generation, compilation technology, real-time systems, operating systems, object-oriented programming, network protocols, software development processes.

