УДК 004.832.3

ПРОВЕРКА СВОЙСТВА КО-НАБЛЮДАЕМОСТИ ФОРМАЛЬНОГО ЯЗЫКА ПРИ ПОМОЩИ ЛОГИЧЕСКОГО ВЫВОДА

А.В. Давыдов

Институт динамики систем и теории управления им. В.М. Матросова СО РАН

Россия, 664033, Иркутск, Лермонтова ул., 134

E-mail: artem@icc.ru

Н.В. Нагул

Институт динамики систем и теории управления им. В.М. Матросова СО РАН

Россия, 664033, Иркутск, Лермонтова ул., 134

E-mail: sapling@icc.ru

А.А. Ларионов

Институт динамики систем и теории управления им. В.М. Матросова СО РАН

Россия, 664033, Иркутск, Лермонтова ул., 134

E-mail: bootfrost@zoho.com

Ключевые слова: дискретно-событийные системы, автоматическое доказательство теорем, автономные необитаемые подводные аппараты.

Аннотация: В докладе рассматривается применение логического исчисления позитивно-образованных формул как средства формализации и решения различных задач супервизорного управления дискретно-событийными системами. На примере дискретно-событийной модели, описывающей переключение режимов работы автономного подводного необитаемого аппарата, действующего в связке лидер-ведомый при выполнении групповой миссии, представлен алгоритм для проверки свойства ко-наблюдаемости языка спецификации. Ко-наблюдаемость языка спецификации является одним из условий существования децентрализованного супервизора, как правило, необходимого для выполнения задач группового управления.

1. Введение

Дискретно-событийные системы (ДСС) представляют собой системы, изменения состояний которых являются результатом возникновения дискретных событий. Некоторые последовательности событий являются нежелательными, что описывается так называемой спецификацией на поведение системы. Автоматное представление ДСС как генератора формального языка является предметом теории супервизорного управления (ТСУ) [1]. Под генератором понимается структура $\mathcal{G} = (Q, \Sigma, \delta, q_0, Q_m)$, где Q – множество состояний q; Σ – множество событий; δ: $\Sigma \times Q \to Q$ – функция

переходов; $q_0 \in Q$ – начальное состояние; $Q_m \subset Q$ – множество маркированных состояний. Супервизор представляет собой средство, позволяющее избежать генерации нежелательных слов, таким образом обеспечивая выполнение спецификации, также описываемой формальным языком. Несмотря на интенсивные исследования с 1980-х годов, в ТСУ все еще остается много открытых проблем, особенно в области частично наблюдаемых и децентрализованных систем, прежде всего из-за вычислительной сложности алгоритмов построения супервизоров.

Для формализации и решения различных задач управления ДСС в рамках ТСУ может успешно применяться оригинальное первопорядковое логическое исчисление позитивно-образованных формул (ПОФ). Исчисление ПОФ разработано в [2, 3], а его дальнейшее развитие можно найти в [4]. Данное исчисление естественным образом подходит для применения в задачах управления динамическими системами благодаря его особенностям, таким как модифицируемость семантики (немонотонная, временна́я и т.д.) и возможность построения конструктивных выводов формул достаточно широкого класса, существенно шире хорновских, используемых в языке Пролог. Чтобы представить основанный на исчислении ПОФ подход к решению задач ТСУ, ниже будет рассмотрен вопрос проверки одного из условий существования децентрализованного супервизора.

2. Проверка ко-наблюдаемости языка спецификации при помощи ПОФ

Рассмотрим ДСС в автоматной форме, в общих чертах описывающую переключение режимов работы автономного необитаемого подводного аппарата (АНПА), выполняющего в составе группы АНПА некоторые заранее запланированные миссии («рис.1»). Предполагается, что группа организована по схеме «лидер-ведомый». Здесь $\Sigma = \{a, b, c, d, e, f\}$, где a – «получена команда на сбор», b – «получена временная роль лидера», c – «миссия выполнена», d – «получена команда ожидания», e – «получены новые параметры миссии», f – «получена команда запуска новой миссии». Состояния из множества $Q = \{LF, PF, G, R, S\}$ соответствуют режимам работы АНПА, движущемуся в некоторой формации. Здесь LF («следование за лидером») и PF («движение по заданной траектории») являются стандартными режимами для ведомого и лидера, соответственно. G соответствуют режиму сбора, который включается, когда завершаются все планы лидера или когда ведомый получает соответствующую команду. R – режим ожидания: ведомые находятся в режиме ожидания, а лидер перенастраивает миссию и ждет поступления очередных команд (с поддерживающего судна, от координатора группы и т.д.). S – режим, который соответствует перемещению АНПА в другую локацию, где начинается новая миссия. Рассматривая события как символы алфавита, $\mathcal{G}$ генерирует формальный язык $L(\mathcal{G}) = \{\overline{[(a + bc)(e + de)f]^*}\}$. Здесь и далее, $\overline{L}$ и L^* – префиксное замыкание и замыкание Клини языка L, соответственно.

ТСУ предполагает возможность управления возникновением событий с целью ограничения функционирования системы для удовлетворения определенных требований, называемых *спецификацией* на поведение системы. При этом множество управляемых событий Σ_c обычно отличается от Σ, $\Sigma_c \subset \Sigma$. Средство управления $\mathcal{G}$ называется *супервизором*. Супервизор разрешает или запрещает возникновение

управляемых событий, однако, в общем случае, не имеет возможности форсировать наступление события.

В случае, если централизованное управление неосуществимо по той или иной причине, применяется децентрализованное супервизорное управление. Предполагается, что имеется набор локальных супервизоров, каждый из которых наблюдает свое подмножество событий $\Sigma_{i,o}$ из Σ и управляет своим подмножеством $\Sigma_{i,c}$ из Σ_c. Действуя совместно, они должны гарантировать выполнение спецификации. Децентрализованное супервизорное управление использует более сложные алгоритмы построения супервизоров и требует проверки условий существования управления, известных как *управляемость* и *ко-наблюдаемость* языка, описывающего спецификацию [5]. В настоящем докладе будет рассмотрен вопрос проверки свойства ко-наблюдаемости. Опуская формальности, суть ко-наблюдаемости можно сформулировать следующим образом: если событие приводит к нарушению спецификации, то хотя бы один из супервизоров обладает достаточной информацией, чтобы запретить это событие.

Для ДСС на «рис.1» предполагается, что управление осуществляют два супервизора, один для лидера группы, а другой для ведомого. Пусть $\Sigma_{1,o} = \{a\}$, $\Sigma_{2,o} = \{b, c\}$, $\Sigma_{1,c} = \{a, e\}$, $\Sigma_{2,c} = \{c, d, e\}$. Рассмотрим пример возможных переходов в $\mathcal{G}$. Предположим, что группа состоит из нескольких АНПА с одним лидером. Группа начинает заранее заданную миссию в некоторой акватории. Состояние ведомых – LF, а лидера – PF. В некоторых случаях, например, когда лидер недоступен из-за трудностей со связью, один из ведомых может временно стать лидером группы и изменить свое состояние по событию b. Когда миссия завершена, группа переключается в режим сбора по событиям a и c. В режиме сбора G лидер может получить сообщение с поверхности или иного координатора о необходмости провести некоторые перенастройки и скорректировать планы, в то время как ведомые ждут команды от лидера. Это тонкое место в модели, поскольку и ведомые, и лидер могут получить команду на ожидание и переключиться в состояние R по событию d. После того, как все необходимые настройки миссии сделаны, группа может перейти в другую локацию (состояние меняется на S) и оставаться там в ожидании команды, чтобы начать новую миссию, которая начинается с события f, завершая таким образом цикл.

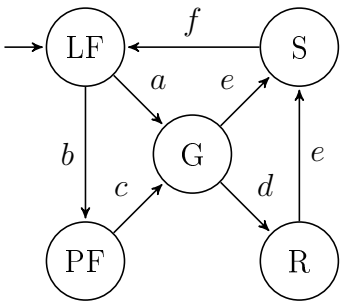

Рис. 1. Генератор $\mathcal{G}$

Анализируя поведение системы, можно отметить, что в состоянии G ведомые не нуждаются в переключении своего состояния в R, потому что они могут получать команды только от лидера. Следовательно, последовательности событий с e после префиксов a и bcd не должны допускаться. Таким образом, язык спецификации есть $K = \{[(ad + bc)ef]^*, \overline{bcd}\}$. Порождающий его автомат изображен на «рис.2».

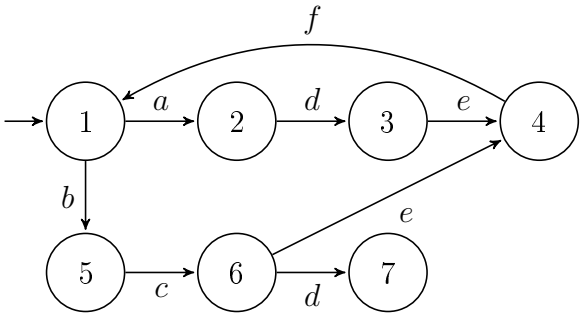

Рис. 2. Автомат H, распознающий язык спецификации K

В то время как в рассматриваемом простом примере найти строку, нарушающую ко-наблюдаемость K, достаточно просто (например, строка bcd), для более сложных задач необходимо делать это автоматически. Формализация ДСС в виде ПОФ, впервые представленная в [6], предоставляет инструмент для проверки ко-наблюдаемости и других важных для ТСУ свойств формальных языков с помощью поиска логического вывода. Чтобы формализовать ДСС на «рис.1», используются следующие предикаты: $E(x)$ интерпретируется как «x - событие», $S(p(x_1, x_2), y, z)$ обозначает «x_1 текущая последовательность событий длины z в состоянии y и x_2 - проекция x_1». Терм p используется для моделирования наблюдаемости текущего события. Функциональный символ «$\cdot$» обозначает конкатенацию строк, а символ «ϵ» обозначает пустую строку.

Генератор описывается формулой, обозначим ее F_G, состоящей из базы $\mathcal{B}$ с атомами $E(a)$, $E(b)$, $E(c)$, $E(d)$, $E(e)$, $E(f)$, $S(p_1(\epsilon, \epsilon), LF, 0)$, $S(p_2(\epsilon, \epsilon), LF, 0)$ и пяти пар вопросов, соответствующих переходам в $\mathcal{G}$, которые выглядят как

$$\forall_{\sigma,\sigma',l}\ E(x), S(p_i(\sigma, \sigma'), M_1, l) - \exists\ S(p_i(\sigma \cdot x, \sigma' \cdot x), M_2, l + 1),$$

$i = \{1, 2\}$, $p_1(\cdot, \cdot)$ соответствует первому супервизору, а $p_2(\cdot, \cdot)$ второму, M_1 это начальное состояние перехода, а M_2 состояние, в которое осуществляется переход. Полная формула F_G получается довольно громоздкой и здесь не приводится. Логический вывод F_G геннерирует язык $L(\mathcal{G})$ в базовом конъюнкте $\mathcal{B}$ в виде первых аргументов атомов S.

ПОФ, соответствующая языку спецификации K, строится аналогично формуле для $L(\mathcal{G})$ по переходам автомата H и далее будет обозначаться F_K. Чтобы проверить ко-наблюдаемость K, F_K параметризуется дополнительным вопросом $\forall_z$ Ф. База остается такой же, как в F_K, а пары вопросов выглядят как

$$\forall_{\sigma,\sigma',l'}\ E(x), S(p_i(\sigma, \sigma'), M_1, l'), l' \leqslant l - \exists\ S(p_i(\sigma \cdot x, \sigma' \cdot x), M_2, l' + 1),$$

$i = \{1, 2\}$. Параметры l и Ф получаются с помощью вывода формулы F^*_G, параметризованной версии F_G, вопросы которой изменяются путем добавления атомов вида $P(\#(\Phi = Q^r_i))$. Здесь Q^r_i являются правыми частями текущего вопроса, копиями с измененными на переменную z вторыми аргументами атомов S. Например, первый вопрос, соответствующий переходу из исходного состояния LF в G, выглядит следующим образом:

$$\forall_{\sigma,\sigma',l}\ E(a), S(p_1(\sigma, \sigma'), LF, l) - \exists\ S(p_1(\sigma \cdot a, \sigma' \cdot a), G, l + 1), \\ P(\#(\Phi = S(p_1(\sigma \cdot a, \sigma' \cdot a), z, l + 1))).$$

Функция # является вычислимой (фактически, выводимой в этом случае), и ее значение определяется подвыводом F_K. # возвращает два значения в качестве возможных вариантов завершения подвывода: константу ok, если подвывод закончился ответом на целевой вопрос $\forall_z$ Ф, или константу $stop$, если подвывод закончился исчерпанием возможных вариантов подстановок. Последнее доказывает выполнимость формулы, что означает, что язык K не обладает свойством ко-наблюдаемости. Дополнительные атомы $l' \leqslant l$ служат ограничением при поиске вывода: они ограничивают машину вывода от поиска дальше, чем длина текущей цепочки событий. Кроме того, дополнительный стоп-вопрос $\forall P(stop)$ добавляется к базе формулы для $\mathcal{G}$ в качестве сигнала остановки для машины вывода. Таким образом, вывод формулы F^*_G с помощью специальной стратегии и поисков подвыводов для определения параметров автоматически приведет к ответу на вопрос о ко-наблюдаемости спецификации K.

3. Заключение

В отличие от известных алгоритмов проверки ко-наблюдаемости языков спецификаций [7], [8], представленный подход позволяет строить ко-наблюдаемые подъязыки данного языка и использовать формализованную информацию об окружающей среде во время вывода. В дальнейших работах подход будет применен не только к проверке условий существования супервизоров, но и к их реализации, в том числе для систем с частичной наблюдаемостью событий. Значительное внимание будет уделено системам, построенным по модульному принципу, т.е. когда поведение системы описывается набором автоматов. Модульная конструкция системы и спецификации предоставляет возможность модульного построения супервизора, и применение ПОФ позволит использовать модульность более эффективно.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (16-29-04238-офи_м).

Список литературы

1. Ramadge P.J., Wonham W.M. Supervisory control of class of discrete event processes // SIAM J. Control and Optimisation. 1987. Vol. 25, No. 1. P. 206-230.
2. Vassilyev S.N. Machine Synthesis of Mathematical Theorems // The Journal of Logic programming. 1990. Vol. 9, No. 2-3. P. 235-266.
3. Васильев С.Н., Жерлов А.К., Федунов Е.А., Федосов Б.Е. Интеллектное управление динамическими системами. М.: Физико-математическая литература, 2000. 352 с.
4. Davydov A.V., Larionov A.A., Cherkashin E.A. On the calculus of positively constructed formulas for automated theorem proving // Automatic Control and Computer Sciences (AC&CS). 2011. Vol. 45, No. 7. P. 402-407.
5. Cassandras C. G., Lafortune S. Introduction to Discrete Event Systems. Springer, 2008.
6. Davydov A., Larionov A., Nagul N. The formal description of discrete-event systems using positively constructed formulas // Proceedings of the 40th International Convention MIPRO. Opatija, Chroatia. CIS - Intelligent Systems. 2017. P. 1161-1165.
7. Rudie K., Willems J. C. The computational complexity of decentralized discrete-event control problems //IEEE Transactions on Automatic Control. 1995. Vol. 40, No. 7. P. 1313-1319.
8. Ricker L., Caillaud B. Mind the gap: Expanding communication options in decentralized discrete-event control // Automatica. 2011. Vol. 47, No. 11, P. 2364-2372.