УДК 004.415.52, 004.434

ПРИМЕНЕНИЕ УСЛОВНЫХ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В ЗАДАЧАХ ВЕРИФИКАЦИИ УПРАВЛЯЮЩИХ АВТОМАТНЫХ ПРОГРАММ

Ф.А. Новиков
Санкт-Петербургский политехнический университет Петра Великого
Россия, 195251, Санкт-Петербург, Политехническая ул., 29
E-mail: fedornovikov51@gmail.com

И.В. Афанасьева
Специальная астрофизическая обсерватория РАН
Россия, 369167, Нижний Архыз
E-mail: riv615@gmail.com

Л.Н. Федорченко
Санкт-Петербургский Федеральный исследовательский центр РАН
Россия, 199178, Санкт-Петербург, 14 Линия В.О., 39
Санкт-Петербургский государственный университет
Россия, 199034, Санкт-Петербург, Университетская наб., 7/9
E-mail: lnf@iias.spb.su

Т.А. Харисова
Санкт-Петербургский политехнический университет Петра Великого
Россия, 195251, Санкт-Петербург, Политехническая ул., 29
E-mail: harisova.ta@edu.spbstu.ru

Ключевые слова: дискретные системы управления, семантика автоматной программы, условные регулярные выражения, верификация автоматной программы.

Аннотация: В данной работе для анализа дискретных систем управления применяется формальный подход, основанный на кооперативно взаимодействующих автоматных объектах, определенных на языке CIAO (Cooperative Interaction Automata Objects). Поведение всей системы складывается из поведения автоматных объектов, взаимодействующих через определенные интерфейсы. Поведение каждого объекта описывается специальным автоматом, представленным графом переходов состояний, переходы которого нагружены действиями со сторожевыми условиями. Для целей верификации системы управления элементарные действия рассматриваются как символы конечного алфавита, последовательности действий (протоколы выполнения) образуют слова в этом алфавите, все множество последовательностей действий составляет некоторый язык (семантику системы). Тогда требования к системе (то есть спецификация) представляют собой утверждения о структуре слов этого языка. Рассматривается случай, когда спецификация задается условными регулярными выражениями, определяющими допустимые (недопустимые) последовательности событий (действий). В этом случае задача верификации системы сводится к проверке того, что все слова семантики системы принадлежат языку, заданному условными регулярными выражениями спецификации системы.

1. Введение

Проверка правильности управляющей программы – проверка ее соответствия ожидаемому поведению – называется валидацией. Свойство правильности, как таковое, нельзя проверить в общем случае, но можно проверить соответствие программы формальным требованиям, предъявляемым к правильной программе. Набор таких требований называется спецификацией, а проверка соответствия программы ее спецификации – верификацией. В данном случае под валидацией понимается проверка правильности спецификации [1].

Язык спецификации взаимодействующих автоматных объектов CIAO (Cooperative Interaction Automata Objects) предназначен для описания поведения распределенных и параллельных систем, управляемых событиями [2]. Суть событийно-управляемых систем (или дискретных реагирующих систем) заключается в поддержании взаимодействия с окружением – система отвечает на внешние события определенными действиями. В отличие от программ преобразования данных (или трансформационных программ), для верификации реагирующих систем недостаточно проверить соотношения между входом и выходом после прихода системы в заключительное состояние: реагирующие системы в большинстве случаев функционируют неопределенно долго, и для правильности их работы важно обеспечить соответствие между последовательностью входных событий и последовательностью действий системы [3].

При описании управляющей системы на языке CIAO поведение каждого объекта может быть описано автоматом – графом переходов состояний. Поведение всей системы складывается из поведения автоматных объектов, взаимодействующих через определенные интерфейсы. Для целей описания семантики и верификации, каждой системе управления, описанной на языке CIAO, может быть сопоставлен единый семантический граф, как показано в разделе 2. В такой интерпретации семантика системы – это множество всех возможных протоколов выполнения (последовательностей элементарных действий), то есть множество всех возможных путей в семантическом графе, исходящих из начального состояния.

Считая элементарные действия буквами некоторого алфавита, семантический граф можно рассматривать как граф-источник некоторого формального языка, все слова которого читаются как пути в семантическом графе. Тогда для описания требований к системе, то есть для задания спецификации, можно использовать средства описания формальных языков. В данной работе рассматривается известное средство описания языков – регулярные выражения с условиями [4], рассмотренные в разделе 3.

Имея семантику программы, заданную в форме семантического графа и спецификацию программы, заданную в форме условного регулярного выражения, можно построить алгоритм автоматической верификации программы, как показано в заключении. Другими словами, в некоторых случаях можно проверить правильность программы управления математическими средствами, без тестирования.

2. Построение семантического графа

Семантический граф для системы взаимодействующих автоматных объектов на языке CIAO строится автоматически с помощью несложного алгоритма. Идею этого алгоритма легко понять на примере, и многочисленные технические подробности не нуждаются в формальном описании. Рассмотрим пример программы из статьи [5]. Рассматривается взаимодействие двух объектов, именуемых Производитель и Потребитель, которые оба обладают «свободой воли», то есть работают асинхронно по своим алгоритмам, и взаимодействуют через указанные интерфейсы (рис. 1).

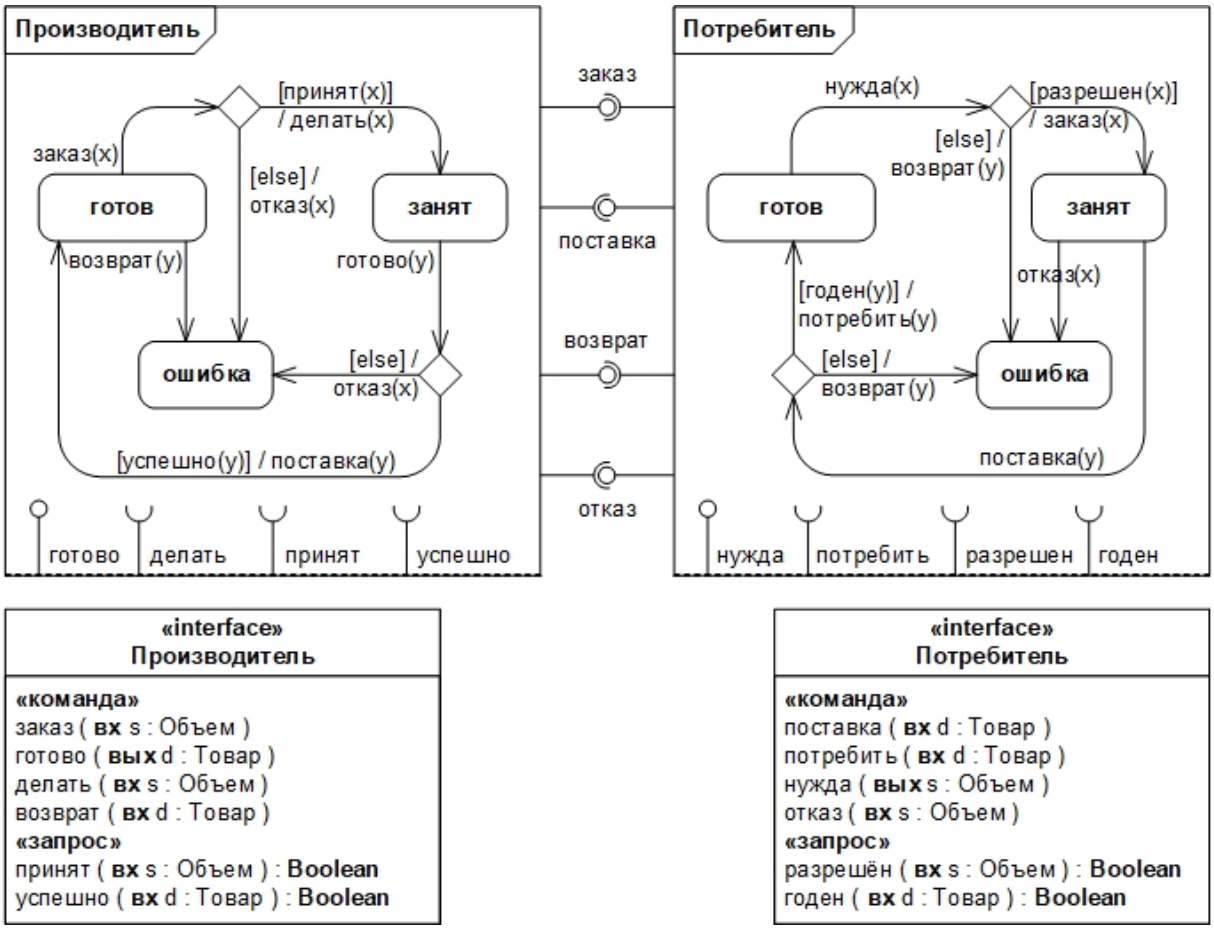

Рис. 1. Взаимодействие между производителем и потребителем с учетом возможных исключительных ситуаций.

Исследуя возможные последовательности событий/действий, получаем семантический граф, представленный на рис. 2.

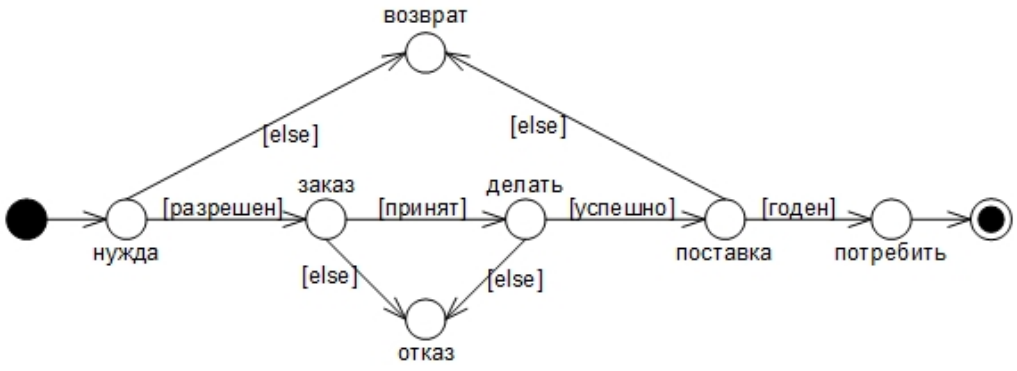

Рис. 2. Семантический граф взаимодействия производителя и потребителя с учетом возможных исключительных ситуаций.

3. Условные регулярные выражения

Идею формализации спецификаций также легко пояснить на примере.

Предположим, что к данной системе предъявляется разумное неформальное требование: можно и нужно потреблять только те товары, которые разрешены к потреблению и которые пригодны для потребления. На языке семантических графов это означает, что все пути в семантическом графе, которые содержат действие «потребить» должны обязательно пройти перед ним сторожевые условия «[разрешен]» и «[годен]».

Такое свойство путей в семантическом графе можно записать с помощью условных регулярных выражений. Ограничим язык регулярных выражений операциями альтернации, сцепления и итерации Цейтина (обобщенная итерация Клини [4]). Обозначим действие «потребить» через p, условие «[разрешен]» через r и условие «[годен]» через g. Пусть также выражение $[\hat{}a]$ означает класс любых элементов, за исключением элемента a. Тогда формально требуемое записывается следующей простой формулой:

(1) $$[\hat{}p]\#\, r\, [\hat{}p]\#\, g\, [\hat{}p]\#\, p.$$

В формуле (1) используется только одна из доступных операций – итерация Цейтина, которая обозначается «# » [6].

5. Заключение

Алгоритмы синтаксического анализа [4] позволяют эффективно проверить, что множество слов, задаваемых семантическим графом на рис. 2, содержится (или не содержится) в регулярном языке, заданном выражением (1). В данном примере совершенно очевидно, что существует путь, который начинается с действия «нужда», заканчивается действием «потребить» и при этом проходит через условия «[разрешен]» и «[годен]» и не существует пути, который начинается с действия «нужда», заканчивается действием «потребить» и при этом не проходит через условия «[разрешен]» и «[годен]».

В заключение следует подчеркнуть, что предложенный метод автоматической верификации программ является простым и эффективным, но не является универсальным и легко применимым. Если программа задана в форме взаимодействующих автоматных объектов, а спецификация – в форме условных регулярных выражений, то предложенный метод позволяет автоматически, без тестирования, математически строго проверить соответствие программы требованиям. Но для появления взаимодействующих автоматных объектов необходим программист, владеющий искусством автоматного программирования, и необходим постановщик задач, способный сформулировать неформальные требования к системе на языке условных регулярных выражений. Это процессы являются творческими и пока требуют участия человека. Однако существуют и имеют особое значение так называемые «ответственные приложения», в которых тестирование невозможно или нецелесообразно, и методы математически строгой верификации имеют неоценимое значение.

Список литературы

1. Шалыто А.А. Валидация автоматных спецификаций // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 2. С. 436-438.
2. Новиков Ф.А., Афанасьева И.В. Кооперативное взаимодействие автоматных объектов // Информационно-управляющие системы. 2016. № 6. С. 50-63.

3. Афанасьева И.В., Новиков Ф.А., Федорченко Л.Н. Методика построения событийно-управляемых программных систем с использованием языка спецификации CIAO // Труды СПИИРАН. 2020. Т. 19, № 3. С. 481-514.
4. Aho A.V., Lam M.S., Sethi R., Ullman J.D. Compilers: principles, techniques, and tools / 2nd ed. Boston: Pearson/Addison-Wesley, 2007.
5. Афанасьева И.В., Новиков Ф.А., Федорченко Л.Н. Верификация событийно-управляемых программных систем с использованием языка спецификации взаимодействующих автоматных объектов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 4. С. 750–756 (на англ. яз.).
6. Fedorchenko L., Baranov S. Equivalent Transformations and Regularization in Context-Free Grammars // Cybernetics and Information Technologies. 2015. Vol. 14, No. 4. P. 29-44.