

МЕТОД ФОРМАЛЬНОЙ ВЕРИФИКАЦИИ ПРОГРАММ ДЛЯ ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ «МАШИНА ПОСТА»

А. В. Лямин^а, канд. техн. наук, доцент, lyamin@mail.ifmo.ru

^аСанкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Кронверкский пр., 49, Санкт-Петербург, 197101, РФ

Введение: формальная верификация программ в символьном виде позволяет в автоматическом режиме осуществлять оценку практических навыков обучающихся при наличии неразрешимого множества верных ответов, что существенно облегчает разработку программных компонентов, используемых для оценки. **Цель:** разработка нового метода формальной верификации программ в символьном виде для виртуальной лаборатории «Машина Поста». **Результаты:** разработан новый метод формальной верификации программ в символьном виде для виртуальной лаборатории «Машина Поста». Сформулированы основные принципы и примеры формирования программ для машины Поста и методы их символьной формализации. Рассмотрены различные задачи использования математических операций с представленными на ленте машины числами в унарном коде, в частности: прибавление единицы к числу, сложение и умножение двух чисел, остановка работы программы в определенной конфигурации ленты машины, переход к новой конфигурации машины от первоначально заданной. Рассмотренных задач достаточно для формирования у обучающихся базовых практических навыков алгоритмизации, а символьное представление решения таких задач позволит упростить разработку оценивающих средств виртуальной лаборатории «Машина Поста» и осуществлять контроль формирования данных навыков в автоматическом режиме. Представлена концепция внедрения разработанного метода верификации программ в уже существующие обучающие информационные системы. **Практическая значимость:** разработанный метод формальной верификации программ для виртуальной лаборатории «Машина Поста» позволяет осуществлять оценку практических навыков алгоритмизации задач у обучающихся и упростить разработку виртуальных лабораторий путем уменьшения затрат на формирование тестовых наборов данных для проведения оценивающих мероприятий.

Ключевые слова — информационные системы, электронное обучение, виртуальные лаборатории, формальная верификация программ.

Цитирование: Лямин А. В. Метод формальной верификации программ для виртуальной лаборатории «Машина Поста» // Информационно-управляющие системы. 2018. № 2. С. 104–114. doi:10.15217/issn1684-8853.2018.2.104

Citation: Lyamin A. V. Method of Formal Program Verification for Post Machine Virtual Laboratory. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2018, no. 2, pp. 104–114 (In Russian). doi:10.15217/issn1684-8853.2018.2.104

Введение

Формирование практических навыков у обучающихся в ходе образовательного процесса всегда было одной из первоочередных задач образовательных организаций. В связи с развитием современного образования, в частности электронного обучения, широкое распространение получили виртуальные лаборатории [1, 2], представляющие собой программные компоненты электронных информационно-образовательных сред, доступные для обучающихся. Развитие современного образования способствует созданию большого числа массовых открытых онлайн-курсов [3–5], реализующих обучение в электронной среде. В этом случае использование виртуальных лабораторий является наиболее перспективным методом формирования практических навыков обучающихся и их оценивания.

С развитием технологий появилось множество различных типов виртуальных лабораторий, которые могут быть разделены на три классификационные группы в соответствии с их основными характеристиками.

Первая классификационная группа определяет типы лабораторий в зависимости от их

программной реализации. Автономные виртуальные лаборатории [6] представляют собой специализированное программное обеспечение, устанавливаемое на пользовательские рабочие станции и не требующее сетевого подключения к сети Интернет. Одним из недостатков такой реализации является невозможность постоянного сохранения данных о выполненных работах. Клиент-серверные виртуальные лаборатории [7] предоставляют доступ обучающимся к лаборатории через веб-браузер и требуют от клиентской станции сетевого подключения, что позволяет осуществлять сбор необходимых сведений об активности обучающихся при выполнении заданий лаборатории в оперативном режиме.

Вторая классификационная группа разделяет виртуальные лаборатории в зависимости от их аппаратной реализации. Использование виртуальных лабораторий с доступом к физическому оборудованию [8] существенно снижает стоимость разработки программного обеспечения, однако требует увеличения расходов на сопровождение. Использование математических моделей при разработке [9] потребует усложнения архитектуры программного обеспечения виртуальной

лаборатории, однако в то же время существенно снизит затраты на покупку и поддержку функционирования оборудования.

Третья классификационная группа включает виртуальные лаборатории, реализующие возможность автоматической проверки ответов обучающихся [10]. Возможность автоматизированной проверки позволяет с легкостью использовать виртуальные лаборатории в качестве одного из основных инструментов оценивания в массовых открытых онлайн-курсах. В то же время виртуальные лаборатории могут иметь неразрешимое множество правильных ответов, в отличие от стандартных инструментов оценивания, таких как электронные тесты. Для таких заданий правильный ответ не может быть описан конечным множеством или общерекурсивной характеристической функцией, что делает задачу автоматизации проверки таких решений актуальной в рамках развития современного образования. Верификация программ в символьном виде является одним из наиболее перспективных методов решения поставленной задачи. В данной статье описывается созданный метод формальной верификации программ в символьном виде для виртуальной лаборатории «Машина Поста».

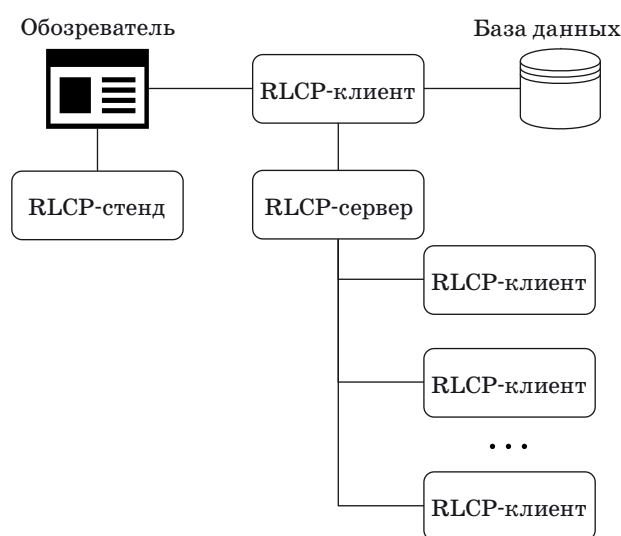
Управление удаленной лабораторией

Один из основных способов верификации результатов прохождения заданий виртуальной лаборатории с неразрешимым множеством правильных ответов представлен верификацией результатов на основе наборов входных и выходных данных. В этом случае ответ обучающегося на представленное ему в рамках виртуальной лаборатории задание содержит описание некоторой системы с установленным набором входов и выходов. Система строится на виртуальном стенде, представляющем собой пользовательскую часть виртуальной лаборатории, встраиваемую в информационно-образовательную среду. На виртуальном стенде отображается представление системы в визуальной форме, а также задание, предназначенное для выполнения. В описанном случае правильный ответ на конкретную поставленную задачу представляется в формате пар эталонных наборов входов и выходов, которые используются для оценки корректности ответа обучающегося после завершения им работы с виртуальной лабораторией. Ответ считается правильным, если реакция системы при проверке на заданных входных данных выдает результат в виде соответствующего ему эталонного набора выходных данных. Для анализа реакции системы, описанной обучающимся в ответе, на заданных эталонных наборах данных использу-

ется виртуальная машина. При этом информационно-образовательная среда обеспечивает организацию взаимодействия виртуального стенда с виртуальной машиной, формирование заданий и эталонных наборов входных и выходных данных, формирование протоколов по результатам действий обучающегося в рамках виртуального стенда и полученных им результатов.

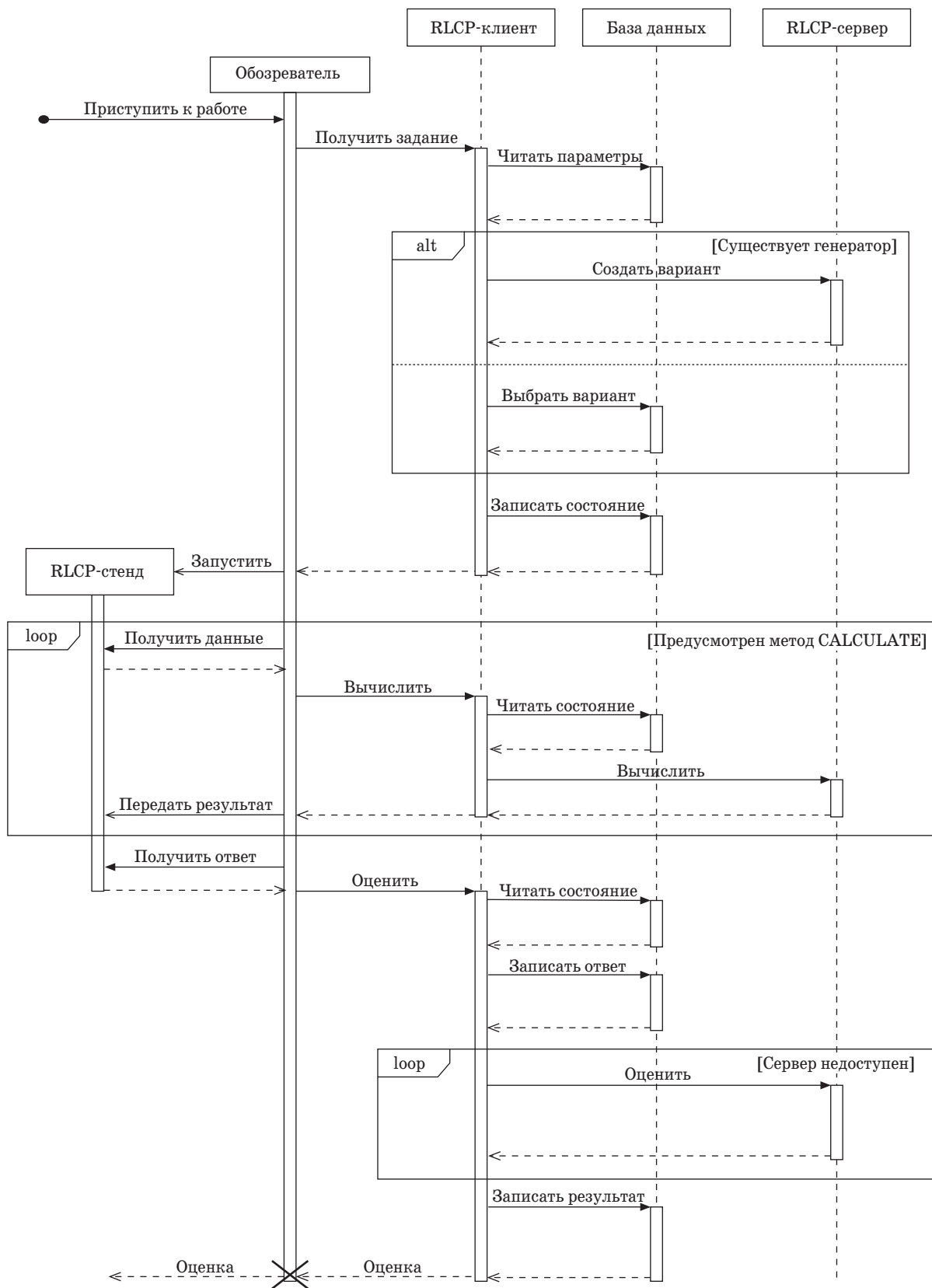
Виртуальные лаборатории представляют собой программные компоненты информационно-образовательной среды, нацеленные на отдельные предметные области. Для унификации механизма доступа к различным виртуальным лабораториям и работы с ними был разработан метод управления удаленной лабораторией, реализованный в протоколе управления удаленной лабораторией RLCP (Remote Laboratory Control Protocol) [11–13]. В протоколе RLCP определяются требования к виртуальному стенду и виртуальной машине, использующимся для построения виртуальных лабораторий; описывается схема взаимодействия между различными компонентами информационно-образовательной среды в процессе формирования задания и его выполнения в целом; описывается механизм проверки результата выполнения заданий виртуальной лаборатории обучающимся. Общая схема построения RLCP-совместимых виртуальных лабораторий приведена на рис. 1.

Виртуальный стенд RLCP-совместимой виртуальной лаборатории называется RLCP-стендом, виртуальная машина — RLCP-сервером. RLCP-клиент представляется в виде контроллера информационно-образовательной среды. Он реали-



■ Рис. 1. Схема построения RLCP-совместимых виртуальных лабораторий

■ Fig. 1. Scheme of building RLCP-compatible virtual laboratories



■ **Рис. 2.** Диаграмма последовательности выполнения лабораторной работы с использованием RLCP-совместимой виртуальной лаборатории
 ■ **Fig. 2.** Sequence diagram of passing an assignment using RLCP-compatible virtual laboratory

зует взаимодействие с обозревателем по протоколу HTTP или HTTPS и с RLCP-сервером с использованием протокола RLCP. Доступ к базе данных выполняется с помощью программного интерфейса RLCP-клиента. Взаимодействие RLCP-стенда с обозревателем проводится за счет программного интерфейса RLCP-стенда.

После взаимодействия обучающегося с интерфейсом информационно-образовательной среды по его инициативе выполняется отправка обозревателем RLCP-клиенту HTTP-запроса на получение задания для виртуальной лаборатории (рис. 2). В RLCP предусмотрен метод генерирования задания виртуальной лаборатории GENERATE [14] на основе некоторого базового шаблона с динамической подстановкой варьирующихся значений. После считывания параметров виртуальной лаборатории RLCP-клиент формирует вариант задания с использованием метода GENERATE или же выбирает готовый вариант из существующего набора, хранящегося в базе данных, в зависимости от наличия генератора заданий для конкретной виртуальной лаборатории. Сформированный вариант задания виртуальной лаборатории фиксируется RLCP-клиентом в базе данных для последующей оценки ответа обучающегося, после чего вариант преобразуется в веб-страницу, содержащую вариант задания и RLCP-стенд, и отправляется обозревателю. Путем взаимодействия с RLCP-стендом обучающийся выполняет задание виртуальной лаборатории. При необходимости обучающийся получает возможность использовать ресурсы RLCP-сервера для выполнения сложных вычислений или предварительной проверки текущего решения работы. Данная функция реализуется в методе RLCP-CALCULATE. Разрешение использовать данный метод должно быть установлено регламентом проведения лабораторной работы. По завершении выполнения задания обучающийся подтверждает готовность решения путем инициализации отправки решения на проверку. В этом случае обозреватель получает данные решения обучающегося из RLCP-стенда и отправляет их RLCP-клиенту в виде HTTP-запроса. Решение оценивается RLCP-сервером после отправки ему RLCP-клиентом с использованием метода CHECK информации из базы данных, необходимой для выполнения процедуры оценивания работы. В случае невозможности обработки RLCP-сервером информации в конкретный момент, RLCP-клиент продолжает отправление данных до получения ответа от сервера. После завершения обработки результаты оценивания фиксируются в базе данных и становятся доступны обучающемуся.

Любое взаимодействие по протоколу RLCP инициализируется RLCP-клиентом, поэтому данный протокол является асимметричным. После-

довательность взаимодействия можно представить в следующем виде:

- 1) установление TCP-соединения;
- 2) отправка запроса RLCP-клиентом (запрос содержит строку состояния с названием метода RLCP, поля заголовка, пустую строку, тело запроса);
- 3) отправка ответа RLCP-сервером;
- 4) разрыв TCP-соединения.

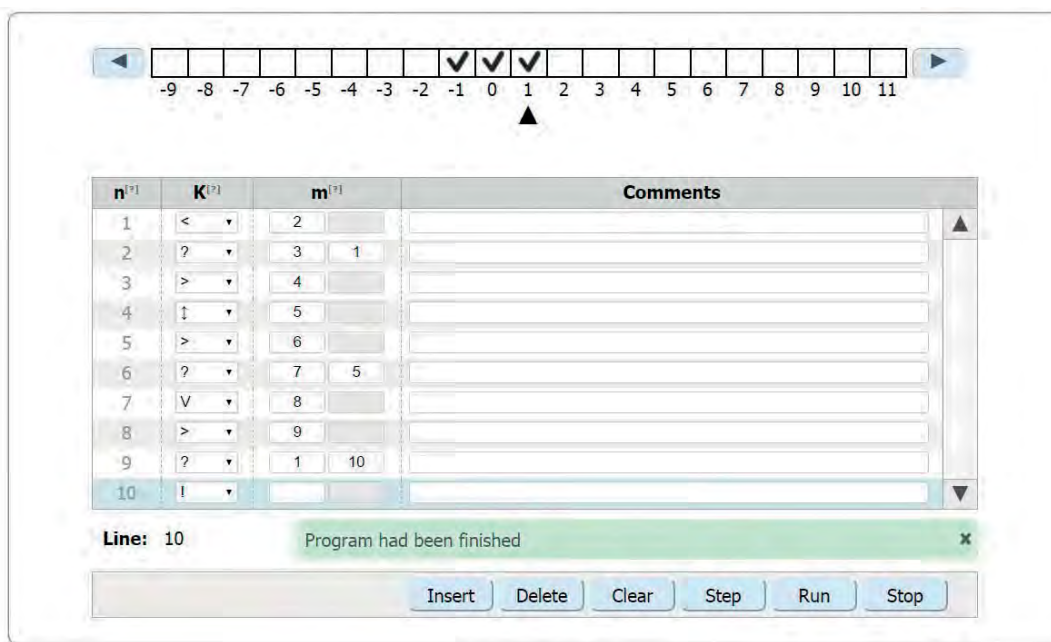
Виртуальная лаборатория «Машина Поста»

Абстрактные алгоритмические машины являются одним из основных инструментов формирования базовых навыков решения алгоритмических задач в информатике. Виртуальные лаборатории, моделирующие такие машины, позволяют обучающимся сформировать необходимые навыки алгоритмизации [15] задач путем использования виртуального стенда лаборатории, визуализирующего конкретную абстрактную машину. Машина Поста [16] является одной из абстрактных алгоритмических машин, введенных для формализации понятия «алгоритм». Данный тип абстрактных машин удобен для формирования базовых навыков алгоритмизации и оценивания навыков анализа алгоритмов, поскольку использование машины Поста не требует знания сложных языков программирования и подходит для применения обучающимися с любым уровнем знаний. В автоматизированной системе управления процессом электронного обучения AcademicNT была разработана и размещена в системе RLCP-совместимая виртуальная лаборатория, моделирующая работу машины Поста [13].

RLCP-стенд лаборатории включает несколько основных компонент: бесконечную ленту, на которой размещены ячейки для хранения информации; редактор команд машины; область для вывода сопутствующей информации, в том числе номера текущей выполняемой команды и системных сообщений; элементы управления виртуальным стендом. В случае возникновения ошибок или успешного завершения выполнения программы виртуальный стенд выдает соответствующие сообщения. Интерфейс виртуального стенда лаборатории «Машина Поста» представлен на рис. 3.

Для обучающегося при взаимодействии с виртуальным стендом доступны следующие действия:

- установка метки в неотмеченную ячейку;
- снятие метки с отмеченной ячейки;
- перемещение ленты машины вправо и влево;



■ Рис. 3. RLCP-стенд виртуальной лаборатории «Машина Поста»
 ■ Fig. 3. RLCP-stand of Post machine virtual laboratory

- Перечень команд машины Поста
- List of Post machine commands

Команда	Описание
$n \rightarrow m$	Сдвиг каретки вправо, переход к команде с номером m
$n \leftarrow m$	Сдвиг каретки влево, переход к команде с номером m
$n \vee m$	Установка метки, переход к команде с номером m
$n \times m$	Снятие метки, переход к команде с номером m
$n ? m_1, m_2$	Если ячейка не отмечена — переход к команде с номером m_1 , иначе — переход к команде с номером m_2
$n !$	Команда остановки

— ввод / редактирование команды nKm в редакторе команд, где n — номер текущей команды, K — операция машины Поста (описание полного списка операций приведено в таблице), m — номер следующей команды;

— выполнение управляющих операций: добавление / удаление команд, снятие меток со всех отмеченных ячеек на ленте, запуск программы в пошаговом режиме, запуск программы на выполнение, остановка режима пошагового выполнения.

Метод формальной верификации программ

Использование виртуальных лабораторий в образовательном процессе позволяет обучающимся сформировать необходимые в конкретной предметной области практические навыки, а также служат инструментом их оценивания. Однако множество правильных ответов на задания виртуальных лабораторий в подавляющем большинстве случаев является алгоритмически неразрешимым [17], что усложняет процесс разработки соответствующего программного и методического обеспечения. Обучающийся в ходе выполнения задания виртуальной лаборатории описывает определенную систему, имеющую множество входов и выходов. Таким образом, одним из способов верификации правильности результатов выполнения работы является формирование эталонных наборов входных и выходных данных, для которых осуществляется проверка функционирования описанной обучающимся системы. Однако невозможно сформировать множество наборов эталонных данных, поскольку в большинстве случаев данное множество также является неразрешимым, а ограничение оценивания на наиболее характерных наборах данных не гарантирует правильность работы системы на других наборах. Для решения поставленной проблемы может быть применен метод символического выполнения [18–20] как один из наиболее эффективных методов оценки в указанном случае.

В рамках данной работы был разработан метод формальной верификации программ в сим-

вольном виде для виртуальной лаборатории «Машина Поста». Машина Поста представляет собой простейшую алгоритмическую машину и имеет бинарный алфавит. Как было описано выше, виртуальный стенд лаборатории «Машина Поста» имеет бесконечную в обе стороны ленту, а также каретку, указывающую на текущую обозреваемую ячейку ленты. Ячейки ленты могут находиться в двух состояниях — быть пустыми (0) или содержать метку (1), таким образом, общее состояние ленты определяется набором значений (словом из нулей и единиц), содержащихся на ленте в конкретный момент времени. В начальный момент времени только конечное количество ячеек содержит метку, остальные остаются пустыми. Язык, описывающий состояния ленты машины Поста, можно определить регулярным выражением $\varepsilon + 1 + 1(0 + 1)^*1$, где ε — пустая цепочка. Для полного описания состояния машины необходимо ввести состояние каретки, указывающей на определенную ячейку. Таким образом, состояние машины определяется тройкой значений $\alpha_1\Lambda\alpha_2$, где α_1 — слово, определяющее состояние фрагмента ленты, расположенного слева от каретки; Λ — символ, обозначающий положение каретки; α_2 — слово, определяющее состояние фрагмента ленты, образованного обозреваемой ячейкой и ячейками, находящимися справа от нее. Стандартная конфигурация машины представляет собой конфигурацию, при которой каретка обозревает крайнюю слева отмеченную ячейку — $\Lambda\alpha$, где α — слово, определяющее состояние ленты.

В соответствии с обозначенным выше перечнем команд машины Поста, список команд состоит из следующих: $i > j$, $i < j$, $i \vee j$, $i \times j$, $i ? j_1 j_2$, $i !$ — где i — номер текущей выполняемой команды; j, j_1, j_2 — номера команд, которым передается управление после выполнения текущей команды. Общее количество программы F обозначим как $|F|$. Правильная команда для машины Поста не должна содержать ссылки на неопределенные в коде команды. В ходе выполнения программы могут произойти следующие события:

1) машина перейдет к невыполнимой команде (команде установки / снятия метки или неопределенной команде) — выполнение программы завершается, остановка считается безрезультатной;

2) достигается команда остановки — выполнение программы завершается, остановка считается результативной;

3) выполнение программы не завершается, машина работает бесконечно.

Обозначим переход между конфигурациями машины как $K_1 \xrightarrow{i} K_2$, где K_1 — начальная конфигурация машины; K_2 — конечная конфигурация; i — команда, благодаря которой произошел переход. Если конфигурация машины измени-

лась после выполнения программы F , содержащей последовательность команд $S = (i_1, i_2, \dots, i_{n-1})$, где $1 \leq i_j \leq |F|$, $j = 1, 2, \dots, n - 1$, то обозначим это как $K_1 \xrightarrow{S} K_2$. В ходе выполнения программы F

машина перерабатывает слово $\alpha_1\alpha_2$ в слово $\beta_1\beta_2$: $F(\alpha_1\alpha_2) = \beta_1\beta_2$. Рассмотрим несколько примеров программ при условии, что выполнение программы начинается из описанной ранее стандартной конфигурации.

1. *Базовые правила вычислений программ машины Поста в символьном виде.*

Пусть $a \in \Sigma \cup \{\varepsilon\}$, $\alpha, \beta \in \Sigma^*$, $x, y \in \mathbb{N}$, где $\Sigma \in \{0, 1\}$ — двоичный алфавит, ε — пустой символ, тогда:

- | | |
|--|--|
| 1) $a^0 = \varepsilon$ | 9) $\alpha\Lambda\beta \xrightarrow{i < j} \alpha\Lambda\beta$ |
| 2) $a^1 = a$ | 10) $\alpha\Lambda 0\beta \xrightarrow{i \vee j} \alpha\Lambda a\beta$ |
| 3) $\alpha^0 = \varepsilon$ | 11) $\alpha\Lambda 1\beta \xrightarrow{i \times j} \alpha\Lambda 0\beta$ |
| 4) $\alpha^1 = \alpha$ | 12) $\alpha\Lambda 1\beta \xrightarrow{i \vee j} \varepsilon$ |
| 5) $\alpha 0^x = 0^x \alpha = \alpha$ | 13) $\alpha\Lambda 0\beta \xrightarrow{i \times j} \varepsilon$ |
| 6) $a^x a^y = a^y a^x = a^{x+y}$ | 14) $\alpha\Lambda\beta \xrightarrow{i ? j_1, j_2} \alpha\Lambda\beta$ |
| 7) $\alpha^x \alpha^y = \alpha^y \alpha^x = \alpha^{x+y}$ | 15) $\alpha\Lambda\beta \xrightarrow{i !} \alpha\Lambda\beta$ |
| 8) $\alpha\Lambda a\beta \xrightarrow{i > j} \alpha\Lambda\beta$ | |

Пример 1. Рассмотрим программу, добавляющую к числу, записанному на ленте, единицу:

- 1 < 2
- 2 \vee 3
- 3 !

В рассматриваемых примерах числа на ленте представляются в унарном коде — число x будет представлено как $11\dots 1 = 1^x$. В случае размещения на ленте нескольких чисел x_1, x_2, \dots, x_n состояние ленты будет представлено в виде $1^{x_1} 0^{y_1} 1^{x_2} 0^{y_2} \dots 0^{y_{n-1}} 1^{x_n}$, где y_1, y_2, \dots, y_n — некоторые натуральные числа. Начальная конфигурация машины имеет вид $\Lambda 1^x$. После выполнения первой команды конфигурация машины перейдет к виду $\Lambda 0 1^x$. Поскольку вторая команда производит установку метки в обозреваемую ячейку, то конфигурация машины станет равной $\Lambda 1^{x+1}$, после чего происходит результативная остановка выполнения программы. Таким образом, становится возможным описать результат программы в символьном виде и восстановить функцию, определяемую программой.

2. *Правила вычисления циклов машины Поста в символьном виде.*

Пример 2. Рассмотрим пример программы, выполняющей сложение двух чисел, т. е. осуществляющей преобразование слова $1^{x_1} 0^{y_1} 1^{x_2}$ в $1^{x_1+x_2}$:

1 > 2	7 > 8	13 > 14
2 ? 3 1	8 ? 7 9	14 ? 13 15
3 < 4	9 < 10	15 < 16
4 X 5	10 V 11	16 V 17
5 < 6	11 < 12	17!
6 ? 13 7	12 ? 11 4	

В представленной программе происходит смещение левой последовательности меток к правой и тем самым формируется результирующая комбинация. Пусть начальная конфигурация машины $\Lambda^1 3^0 3^1 6$, тогда ход выполнения программы можно представить в виде

$\Lambda^1 3^0 3^1 6$	12: $1^2 0^2 \Lambda 0^1 7$	11: $10 \Lambda 0^2 1^8$
1: $1 \Lambda 1^2 0^3 1^6$	11: $1^2 0 \Lambda 0^2 1^7$	12: $10 \Lambda 0^2 1^8$
2: $1 \Lambda 1^2 0^3 1^6$	12: $1^2 0 \Lambda 0^2 1^7$	11: $1 \Lambda 0^3 1^8$
1: $1^2 \Lambda 10^3 1^6$	11: $1^2 \Lambda 0^3 1^7$	12: $1 \Lambda 0^3 1^8$
2: $1^2 \Lambda 10^3 1^6$	12: $1^2 \Lambda 0^3 1^7$	11: $1 \Lambda 10^3 1^8$
1: $1^3 \Lambda 0^3 1^6$	11: $1 \Lambda 10^3 1^7$	12: $1 \Lambda 10^3 1^8$
2: $1^3 \Lambda 0^3 1^6$	12: $1 \Lambda 10^3 1^7$	4: $\Lambda 0^4 1^8$
3: $1^2 \Lambda 10^3 1^6$	4: $1 \Lambda 0^4 1^7$	5: $\Lambda 0^5 1^8$
4: $1^2 \Lambda 0^4 1^6$	5: $\Lambda 10^4 1^7$	6: $\Lambda 0^5 1^8$
5: $1 \Lambda 10^4 1^6$	6: $\Lambda 10^4 1^7$	13: $\Lambda 0^4 1^8$
6: $1 \Lambda 10^4 1^6$	7: $1 \Lambda 0^4 1^7$	14: $\Lambda 0^4 1^8$
7: $1^2 \Lambda 0^4 1^6$	8: $1 \Lambda 0^4 1^7$	13: $\Lambda 0^3 1^8$
8: $1^2 \Lambda 0^4 1^6$	7: $10 \Lambda 0^3 1^7$	14: $\Lambda 0^3 1^8$
7: $1^2 0 \Lambda 0^3 1^6$	8: $10 \Lambda 0^3 1^7$	13: $\Lambda 0^2 1^8$
8: $1^2 0 \Lambda 0^3 1^6$	7: $10^2 \Lambda 0^2 1^7$	14: $\Lambda 0^2 1^8$
7: $1^2 0^2 \Lambda 0^2 1^6$	8: $10^2 \Lambda 0^2 1^7$	13: $\Lambda 0^1 8$
8: $1^2 0^2 \Lambda 0^2 1^6$	7: $10^3 \Lambda 0^1 7$	14: $\Lambda 0^1 8$
7: $1^2 0^3 \Lambda 0^1 6$	8: $10^3 \Lambda 0^1 7$	13: $\Lambda 1^8$
8: $1^2 0^3 \Lambda 0^1 6$	7: $10^4 \Lambda 1^7$	14: $\Lambda 1^8$
7: $1^2 0^4 \Lambda 1^6$	8: $10^4 \Lambda 1^7$	15: $\Lambda 0^1 8$
8: $1^2 0^4 \Lambda 1^6$	9: $10^3 \Lambda 0^1 7$	16: $\Lambda 1^9$
9: $1^2 0^3 \Lambda 0^1 6$	10: $10^3 \Lambda 1^8$	17: $\Lambda 1^9$
10: $1^2 0^3 \Lambda 1^7$	11: $10^2 \Lambda 0^1 8$	
11: $1^2 0^2 \Lambda 0^1 7$	12: $10^2 \Lambda 0^1 8$	

В представленном отображении каждая из строк имеет формат $i: \alpha$, где i — номер текущей завершенной команды, α — конфигурация машины после выполнения команды. В данной программе для выполнения определенных действий используются циклы, в частности:

1: $1 \Lambda 1^2 0^3 1^6$	1: $1^2 \Lambda 10^3 1^6$	1: $1^3 \Lambda 0^3 1^6$
2: $1 \Lambda 1^2 0^3 1^6$	2: $1^2 \Lambda 10^3 1^6$	2: $1^3 \Lambda 0^3 1^6$

В представленном фрагменте повторно выполняются команды 1 и 2, осуществляющие перемещение каретки от начала левой последовательности меток к ее правому концу. Другим примером цикла является повтор команд 11 и 12, осуществляющих перемещение от левого края правой последовательности к правому краю левой, пока в обозреваемой ячейке не будет метки:

11: $1^2 0^2 \Lambda 0^1 7$	11: $1^2 \Lambda 0^3 1^7$
12: $1^2 0^2 \Lambda 0^1 7$	12: $1^2 \Lambda 0^3 1^7$
11: $1^2 0 \Lambda 0^2 1^7$	11: $1 \Lambda 10^3 1^7$
12: $1^2 0 \Lambda 0^2 1^7$	12: $1 \Lambda 10^3 1^7$

Таким образом, можно сформулировать основные характеристики циклов:

1) остановка цикла происходит, если в ходе выполнения цикла уменьшается какая-либо повторяющаяся последовательность из нулей и единиц, записанных на ленте;

2) выполнение цикла продолжается до тех пор, пока уменьшающаяся последовательность не будет исчерпана;

3) цикл не завершается, если ни одна из последовательностей не убывает.

Эти правила позволяют вычислять результат циклов в символьном виде, завершающихся за конечное количество шагов.

Большинство задач для машины Поста предусматривают выполнение сложных операторов — циклов. Обозначим некоторый цикл как S , а также $a \in \Sigma \cup \{\varepsilon\}$, $b, c \in \Sigma$, $b \neq c$, $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_m, \delta_1, \dots, \delta_m \in \Sigma^*$, $x_1, \dots, x_n, y, z_1, \dots, z_m \in \mathbb{N}$, где $f: \mathbb{N} \rightarrow \mathbb{N}$ — возрастающая функция, f^{-1} — обратная функция, $h_1, \dots, h_n, g_1, \dots, g_m: \mathbb{N} \rightarrow \mathbb{Z}$. Если $\Delta^k f = \Delta^k h_1 = \dots = \Delta^k h_n = \Delta^k g_1 = \dots = \Delta^k g_m = 0$, $y - f(k) = 0$, $x_1 + h_1(f^{-1}(y)) \geq 0, \dots, x_n + h_n(f^{-1}(y)) \geq 0$, $z_1 + g_1(f^{-1}(y)) \geq 0, \dots, z_m + g_m(f^{-1}(y)) \geq 0$, то в зависимости от направления движения цикла по командам определим следующие соотношения:

1) движение вправо

$$\begin{aligned} & \alpha_n^{x_n} \beta_n \dots \alpha_1^{x_1} \beta_1 \Lambda b^y c^1 \gamma_1 \delta_1^{z_1} \dots \gamma_m \delta_m^{z_m} \xrightarrow{S} \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(1)} \beta_n \dots \alpha_1^{x_1 + h_1(1)} \beta_1 a^{f(1)} \Lambda \\ & \Lambda b^{y-f(1)} c^1 \gamma_1 \delta_1^{z_1 + g_1(1)} \dots \gamma_m \delta_m^{z_m + g_m(1)} \xrightarrow{S} \dots \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(k)} \beta_n \dots \alpha_1^{x_1 + h_1(k)} \beta_1 a^{f(k)} \Lambda \\ & \Lambda b^{y-f(k)} c^1 \gamma_1 \delta_1^{z_1 + g_1(k)} \dots \gamma_m \delta_m^{z_m + g_m(k)} \xrightarrow{S} \dots \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(f^{-1}(y))} \beta_n \dots \alpha_1^{x_1 + h_1(f^{-1}(y))} \beta_1 a^y \Lambda \\ & \Lambda c^1 \gamma_1 \delta_1^{z_1 + g_1(f^{-1}(y))} \dots \gamma_m \delta_m^{z_m + g_m(f^{-1}(y))}; \end{aligned}$$

2) движение влево

$$\begin{aligned} & \alpha_n^{x_n} \beta_n \dots \alpha_1^{x_1} \beta_1 c^1 b^{x-1} \Lambda b^1 \gamma_1 \delta_1^{z_1} \dots \gamma_m \delta_m^{z_m} \xrightarrow{S} \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(1)} \beta_n \dots \alpha_1^{x_1 + h_1(1)} \beta_1 c^1 b^{y-1-f(1)} \Lambda \\ & \Lambda b^1 a^{f(1)} \gamma_1 \delta_1^{z_1 + g_1(1)} \dots \gamma_m \delta_m^{z_m + g_m(1)} \xrightarrow{S} \dots \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(k)} \beta_n \dots \alpha_1^{x_1 + h_1(k)} \beta_1 c^1 b^{y-1-f(k)} \Lambda \\ & \Lambda b^1 a^{f(k)} \gamma_1 \delta_1^{z_1 + g_1(k)} \dots \gamma_m \delta_m^{z_m + g_m(k)} \xrightarrow{S} \dots \\ & \xrightarrow{S} \alpha_n^{x_n + h_n(f^{-1}(y))} \beta_n \dots \alpha_1^{x_1 + h_1(f^{-1}(y))} \beta_1 \Lambda \\ & \Lambda c^1 a^y \gamma_1 \delta_1^{z_1 + g_1(f^{-1}(y))} \dots \gamma_m \delta_m^{z_m + g_m(f^{-1}(y))}. \end{aligned}$$

В случае если $f(k) = k$ и $\alpha_1 = \dots = \alpha_n = \beta_1 = \dots = \beta_n = \gamma_1 = \dots = \gamma_n = \delta_1 = \dots = \delta_n = \varepsilon$, обозначенные выше соотношения представляются в виде

$$\begin{aligned} \Lambda b^y c^1 \xrightarrow{S} a^1 \Lambda b^{y-1} c^1 \xrightarrow{S} \dots \xrightarrow{S} a^{x+y} \Lambda c^1; \\ c^1 b^{y-1} \Lambda b^1 \xrightarrow{S} c^1 b^{y-2} \Lambda b^1 a^1 \xrightarrow{S} \dots \xrightarrow{S} \Lambda c^1 a^y. \end{aligned}$$

В то же время условие $\Delta^k f = \Delta^k h_1 = \dots = \Delta^k h_n = \Delta^k g_1 = \dots = \Delta^k g_m = 0$ указывает на то, что функции $f, h_1, \dots, h_n, g_1, \dots, g_m$ могут быть интерполированы полиномами, степень которых не превышает $k - 1$.

Пример 3. Вычислим результат выполнения программы, осуществляющей сложение двух чисел, записанных на ленте, при условии, что начальная конфигурация машины равна $\Lambda 1^x 0^y 1^z$:

$\Lambda 1^x 0^y 1^z$	12: $1^{x-1} 0^{y-1} \Lambda 0^1 1^{z+1}$
1: $1^1 \Lambda 1^{x-1} 0^y 1^z$	11: $1^{x-1} 0^{y-2} \Lambda 0^2 1^{z+1}$
2: $1^1 \Lambda 1^{x-1} 0^y 1^z$	12: $1^{x-1} 0^{y-2} \Lambda 0^2 1^{z+1}$
1: $1^2 \Lambda 1^{x-2} 0^y 1^z$	Цикл (11,12):
2: $1^2 \Lambda 1^{x-2} 0^y 1^z$	12: $1^{x-2} \Lambda 10^y 1^{z+1}$
Цикл (1,2):	4: $1^{x-2} \Lambda 0^{y+1} 1^{z+1}$
2: $1^x \Lambda 0^y 1^z$	5: $1^{x-3} \Lambda 1^{10^y+1} 1^{z+1}$
3: $1^{x-1} \Lambda 1^{10^y} 1^z$	6: $1^{x-3} \Lambda 1^{10^y+1} 1^{z+1}$
4: $1^{x-1} \Lambda 0^{y+1} 1^z$	Цикл (7,8,...,12,4,5,6):
5: $1^{x-2} \Lambda 10^y 1^{z+1}$	6: $\Lambda 0^{y+2} 1^{z+x-1}$
6: $1^{x-2} \Lambda 10^y 1^{z+1}$	13: $\Lambda 0^{y+1} 1^{z+x-1}$
7: $1^{x-1} \Lambda 0^{y+1} 1^z$	14: $\Lambda 0^{y+1} 1^{z+x-1}$
8: $1^{x-1} \Lambda 0^{y+1} 1^z$	13: $\Lambda 0^y 1^{z+x-1}$
7: $1^{x-1} 0^1 \Lambda 0^y 1^z$	14: $\Lambda 0^y 1^{z+x-1}$
8: $1^{x-1} 0^1 \Lambda 0^y 1^z$	Цикл (13,14):
Цикл (7,8):	14: $\Lambda 1^{z+x-1}$
8: $1^{x-1} 0^{y+1} \Lambda 1^z$	15: $\Lambda 0^1 1^{z+x-1}$
9: $1^{x-1} 0^y \Lambda 0^1 1^z$	16: $\Lambda 1^{z+x}$
10: $1^{x-1} 0^y \Lambda 1^{z+1}$	17: $\Lambda 1^{z+x}$
11: $1^{x-1} 0^{y-1} \Lambda 0^1 1^{z+1}$	

В результате символьных вычислений получаем строку 1^{z+x} , что является верным ответом.

Концепция внедрения

С целью внедрить метод в автоматизированные системы управления процессом электронного обучения предлагается описанная далее концепция внедрения метода, включающая анализ исходного кода решения задания виртуальной лаборатории, отправленного обучающимся на проверку. Рассмотренные выше примеры программ для машины Поста позволяют сформировать набор символьных представлений возможных базовых блоков программы, которые являются фиксированными для начальной стандартной конфигурации машины Поста. Сформированные наборы можно внести в базу данных системы, формируя хранилище символьных представлений.

Необходимо учитывать, что в символьном виде могут быть представлены не только результаты выполнения программы, но и сам ее код. Данный факт позволяет осуществлять анализ кода программы без необходимости ее полного запуска с целью верификации. Пример сложения двух чисел, представленных на ленте, содержит два идентичных друг другу цикла, отличающихся между собой исключительно номерами команд:

```
...
7 < 8
8 ? 9 7
...
10 < 11
11 ? 12 10
...
```

В символьном виде указанные циклы можно описать как

```
i < i + 1
i + 1 ? i + 2, i,
```

где i — номер команды. Таким образом, при установленной начальной конфигурации машины Поста и правил символьного выполнения операций и циклов появляется возможность вычислить итоговую конфигурацию машины. Циклы (7,8) и (10,11) выполняют операцию перемещения каретки влево до пустой ячейки и осуществляют переход к следующей команде. Идентифицировав эти циклы в ходе анализа кода и проанализировав программу, можно получить полную информацию о системе, описанной обучающимся в ходе выполнения задания. Для рассматриваемого примера цикла (7,8)

$$1^x \Lambda 0^{y-1} 1^z \xrightarrow{(7,8)} \Lambda 0^1 x 0^{y-1} 1^z.$$

Следовательно, правила для цикла были выполнены и машина осуществила переход в новую конфигурацию.

Заключение

После завершения экспериментального исследования были достигнуты следующие результаты.

1. Классифицированы основные виды виртуальных лабораторий, использующихся в образовательном процессе, определены их достоинства и недостатки.
2. Приведено описание разработанного метода управления удаленной лабораторией RLCP, внедренного и успешно используемого в автоматизированной системе управления процес-

сом электронного обучения Университета ИТМО AcademicNT.

3. Разработан метод формальной верификации программ для виртуальной лаборатории «Машина Поста», позволяющий осуществлять оценивание присланных на проверку заданий

виртуальной лаборатории в символьном виде без использования эталонных наборов входных/выходных данных.

4. Предложена концепция реализации и внедрения метода формальной верификации программ в информационно-образовательные системы.

Литература

1. Аврунин О. Г., Крук О. Я., Носова Т. В., Семенец В. В. Технические аспекты разработки виртуальных лабораторных работ по техническим дисциплинам // Открытое образование. 2008. № 3. С. 11–17.
2. Васильев В. Н., Лисицына Л. С., Лямин А. В. Сетевая технология проведения вступительных испытаний по информатике в режиме on-line // Использование информационно-коммуникационных технологий в процессе оценки качества образования: сб. науч. тр. 2008. С. 55–70.
3. Kay J., Reimann P., Diebold E., Kummerfeld B. MOOCs: So Many Learners, So Much Potential... // IEEE Intelligent Systems. 2013. Vol. 28(3). P. 70–77. doi:10.1109/MIS.2013.66
4. Zang X., Iqbal S., Zhu Y., Riaz M. S., Abbas G., Zhao J. Are MOOCs Advancing as Predicted by IEEE CS 2022 Report? // Intern. Conf. on Proc. of the Systems Informatics, Modelling and Simulation (SIMS). 2016. P. 49–55. doi:10.1109/SIMS.2016.14
5. Stuchlikova L., Kosa A. Massive Open Online Courses — Challenges and Solutions in Engineering Education // IEEE 11th Intern. Conf. on Emerging eLearning Technologies and Applications (ICETA). 2013. P. 359–364. doi:10.1109/ICETA.2013.6674459
6. Djeghloud H., Larakeb M., Bentounsi A. Virtual Labs of Conventional Electric Machines // Intern. Conf. on Interactive Mobile and Computer Aided Learning (IMCL). 2012. P. 52–57. doi:10.1109/IMCL.2012.6396450
7. Magyar Z., Zakova K. Using SciLab for Building of Virtual Lab // 9th Intern. Conf. on Information Technology Based Higher Education and Training (ITHET). 2010. P. 280–283. doi:10.1109/ITHET.2010.5480084
8. Hristov G., Zahariev P., Bencheva N., Ivanov I. Designing the Next Generation of Virtual Learning Environments — Virtual Laboratory with Remote Access to Real Telecommunication Devices // Proc. of the 24th EAEEIE Annual Conf. (EAEEIE). 2013. P. 139–144. doi:10.1109/EAEEIE.2013.6576517
9. Moritz D., Willems C., Goderbauer M., Moeller P., Meinel C. Enhancing a Virtual Security Lab with a Private Cloud Framework // IEEE Intern. Conf. on Teaching, Assessment and Learning for Engineering (TALE). 2013. P. 314–320. doi:10.1109/TALE.2013.6654452
10. Ахи А. А., Станкевич А. С., Шалыто А. А. Автоматические методы модификации решений для тестирования проверяющих программ // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 1(77). С. 81–85.
11. Ефимчик Е. А. Автоматическое построение заданий для проверки навыков выполнения операций над множествами // Новые информационные технологии в образовании: материалы междунар. науч.-практ. конф. 2013. С. 278–281.
12. Lisitsyna L. S., Efimchik E. A., Izgareva S. A. RLCP-compatible Virtual Laboratories with 3d-models and Demonstration Mode: Development and Application in e-learning // Smart Innovation, Systems and Technologies. 2018. Vol. 75. P. 75–81. doi:https://doi.org/10.1007/978-3-319-59451-4_8
13. Efimchik E. A., Cherepovskaya E. N., Lyamin A. V. RLCP-Compatible Virtual Laboratories in Computer Science // Smart Innovation, Systems and Technologies. 2016. P. 303–314. doi:https://doi.org/10.1007/978-3-319-39690-3_27
14. Efimchik E. A., Chezhin M. S., Lyamin A. V., Rusk A. V. Using Automaton Model to Determine the Complexity of Algorithmic Problems for Virtual Laboratories // 9th Intern. Conf. on Application of Information and Communication Technologies (AICT 2015). 2015. P. 541–545. doi:10.1109/ICAICT.2015.7338619
15. Sandya M. Competence Possessed by Engineering Professionals in Regard to the Prevailing Trends and Worldwide Attitudes Towards Engineering Education // IEEE Intern. Conf. on Engineering Education: Innovative Practices and Future Trends (AICERA). 2012. P. 1–6.
16. Post E. L. Finite Combinatory Processes-Formulation 1 // The Journal of Symbolic Logic. 1936. Vol. 1. N 3. P. 103–105.
17. Вашенков О. Е., Лямин А. В. Методика формирования эталонных наборов входных и выходных данных для анализа результатов выполнения заданий с неразрешимым множеством правильных ответов // Научно-технический вестник информационных технологий, механики и оптики. 2009. № 3 (61). С. 99–105.
18. Arifi S. M., Zahi A., Benabbou R. Semantic Similarity based Evaluation for C Programs through the Use of Symbolic Execution // IEEE Global Engineering Education Conf. (EDUCON). 2016. P. 826–833. doi:10.1109/EDUCON.2016.7474648

19. Cheatham T. E., Holloway G. H., Townley J. A. Symbolic Evaluation and the Analysis of Programs // IEEE Transactions on Software Engineering. 1979. Vol. SE-5. N 4. P. 402–417. doi:10.1109/TSE.1979.234207

20. Vidal G. Closed Symbolic Execution for Verifying Program Termination // IEEE 12th Intern. Working Conf. on Source Code Analysis and Manipulation (SCAM). 2012. P. 34–43. doi:10.1109/SCAM.2012.13

UDC 004.023, 004.424

doi:10.15217/issn1684-8853.2018.2.104

Method of Formal Program Verification for Post Machine Virtual Laboratory

Lyamin A. V.^a, PhD, Tech., Associate Professor, lyamin@mail.ifmo.ru

^aSaint-Petersburg National Research University of Information Technologies, Mechanics and Optics, 49, Kronverkskii Pr., 197101, Saint-Petersburg, Russian Federation

Introduction: The symbolic method of formal program verification is an effective way to evaluate students' practical skills automatically, especially for tasks with an unsolvable set of correct answers, which significantly simplifies the development of evaluation software. **Purpose:** Developing a new method of formal program verification using symbolic evaluation approach for the programs written for Post Machine virtual laboratory. **Results:** A new method of formal verification of programs written for Post Machine virtual laboratory using symbolic evaluation approach has been developed. The basic principles and examples of programs for Post Machine as well as methods for their symbolic formalization are discussed. Various tasks of running mathematical operations on numbers presented in unary code were considered, i.e. incrementation, addition, multiplication, break point, transfer to a different machine configuration, etc. The considered tasks are sufficient for the students to form basic practical skills in making algorithms. Symbolic representation of students' answers simplifies the development of evaluation tools of the virtual laboratory, making it possible to monitor the formation of students' practical skills automatically. A concept of embedding the presented method of formal program verification into the available online educational platforms is presented. **Practical relevance:** The developed method of formal verification of programs written for Post Machine is used to evaluate students' practical skills in making algorithms. It can simplify the development of virtual laboratories by reducing the costs of forming test sets used to evaluate the students' answers.

Keywords — Information Systems, e-Learning, Virtual Laboratories, Formal Program Verification.

Citation: Lyamin A. V. Method of Formal Program Verification for Post Machine Virtual Laboratory. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2018, no. 2, pp. 104–114 (In Russian). doi:10.15217/issn1684-8853.2018.2.104

Reference

1. Avrunin O. G., Kruk O. Ia., Nosova T. V., Semenets V. V. Technical Aspects of Development Virtual Laboratory Works for Technical Disciplines. *Otkrytoe obrazovanie* [Open Education], 2008, no. 3, pp. 11–17 (In Russian).
2. Vasil'ev V. N., Lisitsyna L. S., Liamin A. V. Network Technology of Passing Entrance Computer Science Tests in Online. *Sbornik nauchnykh trudov "Ispol'zovanie informatsionno-kommunikatsionnykh tekhnologii v protsesse otsenki kachestva obrazovaniia"* [Digest of Scientific Works "Use of Information and Communication Technologies in the Process of Assessing the Quality of Education"], 2008, pp. 55–70 (In Russian).
3. Kay J., Reimann P., Diebold E., Kummerfeld B. MOOCs: So Many Learners, So Much Potential... *IEEE Intelligent Systems*, 2013, vol. 28(3), pp. 70–77. doi:10.1109/MIS.2013.66
4. Zang X., Iqbal S., Zhu Y., Riaz M. S., Abbas G., Zhao J. Are MOOCs Advancing as Predicted by IEEE CS 2022 Report? *Intern. Conf. on Proc. of the Systems Informatics, Modelling and Simulation (SIMS)*, 2016, pp. 49–55. doi:10.1109/SIMS.2016.14
5. Stuchlikova L., Kosa A. Massive Open Online Courses — Challenges and Solutions in Engineering Education. *IEEE 11th Intern. Conf. on Emerging eLearning Technologies and Applications (ICETA)*, 2013, pp. 359–364. doi:10.1109/ICETA.2013.6674459
6. Djeghloud H., Larakeb M., Bentounsi A. Virtual Labs of Conventional Electric Machines. *Intern. Conf. on Interactive Mobile and Computer Aided Learning (IMCL)*, 2012, pp. 52–57. doi:10.1109/IMCL.2012.6396450
7. Magyar Z., Zakova K. Using SciLab for Building of Virtual Lab. *9th Intern. Conf. on Information Technology Based Higher Education and Training (ITHET)*, 2010, pp. 280–283. doi:10.1109/ITHET.2010.5480084
8. Hristov G., Zahariev P., Bencheva N., Ivanov I. Designing the Next Generation of Virtual Learning Environments — Virtual Laboratory with Remote Access to Real Telecommunication Devices. *Proc. of the 24th EAEEIE Annual Conf. (EAEEIE)*, 2013, pp. 139–144. doi:10.1109/EAEEIE.2013.6576517
9. Moritz D., Willems C., Goderbauer M., Moeller P., Meinel C. Enhancing a Virtual Security Lab with a Private Cloud Framework. *IEEE Intern. Conf. on Teaching, Assessment and Learning for Engineering (TALE)*, 2013, pp. 314–320. doi:10.1109/TALE.2013.6654452
10. Ahi A. A., Stankevich A. S., Shalyto A. A. Automatic Solution Modifications for Testing Solution Checkers. *Nauchno-tehnicheskij vestnik informacionnykh tekhnologij, mehaniki i optiki*, 2012, no. 1(77), pp. 81–85 (In Russian).
11. Efimchik E. A. Automatic Creation of Tasks for Assessing Skills of Performing Operations on Sets. *Materialy mezhdunarodnoj nauchno-prakticheskoy konferencii "Novye informacionnye tekhnologii v obrazovanii"* [Proc. of the Intern. Scientific and Practical Conf. on the New Educational Information Technologies], 2013, pp. 278–281 (In Russian).
12. Lisitsyna L. S., Efimchik E. A., Izgareva S. A. RLCP-compatible Virtual Laboratories with 3d-models and Demonstration Mode: Development and Application in e-learning. *Smart Innovation, Systems and Technologies*, 2018, vol. 75, pp. 75–81. doi:https://doi.org/10.1007/978-3-319-59451-4_8
13. Efimchik E. A., Cherepovskaya E. N., Lyamin A. V. RLCP-Compatible Virtual Laboratories in Computer Science. *Smart Innovation, Systems and Technologies*, 2016, pp. 303–314. doi:https://doi.org/10.1007/978-3-319-39690-3_27
14. Efimchik E. A., Chezhin M. S., Lyamin A. V., Rusak A. V. Using Automaton Model to Determine the Complexity of Algorithmic Problems for Virtual Laboratories. *9th Intern. Conf. on Application of Information and Communication Technologies (AICT 2015)*, 2015, pp. 541–545. doi:10.1109/ICAICT.2015.7338619
15. Sandya M. Competence Possessed by Engineering Professionals in Regard to the Prevailing Trends and Worldwide Attitudes Towards Engineering Education. *IEEE Intern.*

- Conf. on Engineering Education: Innovative Practices and Future Trends (AICERA)*, 2012, pp. 1–6.
16. Post E. L. Finite Combinatory Processes-Formulation 1. *The Journal of Symbolic Logic*, 1936, vol. 1, no. 3, pp. 103–105.
 17. Vashenkov O. E., Liamin A. V. Creation Technique of Reference Entrance and Target Data Sets for the Evaluation of Tasks with Unsolvable Set of Right Answers. *Nauchno-tekhnicheskii vestnik informatsionnykh tekhnologii, mekhaniki i optiki*, 2009, no. 3 (61), pp. 99–105 (In Russian).
 18. Arifi S. M., Zahi A., Benabbou R. Semantic Similarity based Evaluation for C Programs through the Use of Symbolic Execution. *IEEE Global Engineering Education Conf. (EDUCON)*, 2016, pp. 826–833. doi:10.1109/EDUCON.2016.7474648
 19. Cheatham T. E., Holloway G. H., Townley J. A. Symbolic Evaluation and the Analysis of Programs. *IEEE Transactions on Software Engineering*, 1979, vol. SE-5, no. 4, pp. 402–417. doi:10.1109/TSE.1979.234207
 20. Vidal G. Closed Symbolic Execution for Verifying Program Termination. *IEEE 12th Intern. Working Conf. on Source Code Analysis and Manipulation (SCAM)*, 2012, pp. 34–43. doi:10.1109/SCAM.2012.13

**Научный журнал
«ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ»**

выходит каждые два месяца.

Стоимость годовой подписки (6 номеров) для подписчиков России — 6000 рублей, для подписчиков стран СНГ — 6600 рублей, включая НДС 18%, таможенные и почтовые расходы.

Подписку на печатную версию журнала можно оформить в любом отделении связи по каталогу:

«Роспечать»: № 15385 — полугодовой индекс, а также через посредство подписных агентств:

«Северо-Западное агентство „Прессинформ“»

Санкт-Петербург, тел.: (812) 335-97-51, 337-23-05,

эл. почта: press@crp.spb.ru, zajavka@crp.spb.ru,

сайт: <http://www.pinform.spb.ru>

«МК-Периодика» (РФ + 90 стран)

Москва, тел.: (495) 681-91-37, 681-87-47,

эл. почта: export@periodicals.ru, сайт: <http://www.periodicals.ru>

«Деловая пресса»

Москва, тел.: (495) 962-11-11, эл. почта: podpiska@delpress.ru,

сайт: <http://delpress.ru/contacts.html>

«Коммерсант-Курьер»

Казань, тел.: (843) 291-09-99, 291-09-47, эл. почта: kazan@komcur.ru,

сайт: <http://www.komcur.ru/contacts/kazan/>

«Урал-Пресс» (филиалы в 40 городах РФ)

Сайт: <http://www.ural-press.ru>

«Идея» (Украина)

Сайт: <http://idea.com.ua>

«ВТЛ» (Узбекистан)

Сайт: <http://btl.sk.uz/ru/cat17.html> и др.

На электронную версию нашего журнала (все выпуски, годовая подписка, один выпуск, одна статья)

вы можете подписаться на сайтах НЭБ: <http://elibrary.ru>;

РУКОНТ: <http://www.rucont.ru>; ИВИС: <http://www.ivis.ru/>

Полнотекстовые версии журнала за 2002–2016 гг.

в свободном доступе на сайте журнала (<http://www.i-us.ru>),

НЭБ (<http://www.elibrary.ru>)

и Киберленинки (<http://cyberleninka.ru/>

journal/n/informatsionno-upravlyayuschiesistemy).