

УДК 004.8



И.А. Андрианов, И.А. Гузилов
Вологодский государственный университет

АВТОМАТИЗАЦИЯ ПРОВЕРКИ УЧЕБНЫХ ЗАДАНИЙ ПО ТЕМЕ «МАШИНА ПОСТА»

В статье рассмотрены особенности реализации программы для автоматической проверки решений заданий студентов по теме «Машина Поста». Приведен пример задания, для которого реализована автоматическая проверка. Полученный программный продукт может использоваться в таких учебных курсах, как «Логика и теория алгоритмов», «Построение и анализ алгоритмов».

Машина Поста, теория алгоритмов, обучающие системы, электронное обучение.

Одним из разделов учебного курса «Логика и теория алгоритмов», который преподается в ВоГУ, является изучение универсальных исполнителей – простейших вычислительных устройств, с помощью которых можно реализовать любой алгоритм. Среди таких устройств наиболее известны машина Тьюринга и машина Поста. Обе машины предназначены для доказательства различных утверждений теории алгоритмов. Они были предложены независимо друг от друга в 1936 году английским математиком Аланом Тьюрингом и американским математиком Эмилем Постом [1].

В теории алгоритмов существует так называемый тезис Поста: «всякий алгоритм представим в форме машины Поста». В связи с интуитивностью исходного понятия алгоритмической вычислимости данный тезис (как и аналогичный тезис Черча – Тьюринга) носит характер суждения об этом понятии и его невозможно строго доказать или опровергнуть.

При выполнении практических работ студенты разрабатывают программы для данных машин, решающие указанные алгоритмические задачи. При этом возникает следующая проблема. Проверка решений вручную преподавателем отнимает достаточно много времени. Кроме того, при ручной проверке преподаватель может не заметить ошибку, если она проявляется редко – лишь при некоторых входных данных. В связи с этим актуальна задача автоматизации такой проверки. Для машины Тьюринга такая задача уже была нами решена ранее, в данной же статье описывается автоматизация проверки решений для машины Поста.

Машина Поста состоит из каретки и бесконечной ленты, разделенной на ячейки. Каждая ячейка может быть пустой или помеченной меткой. За один шаг каретка может переместиться на одну позицию вправо или влево либо остаться на месте.

Работа машины определяется программой, состоящей из набора команд, где каждая команда может быть шести различных типов. Машина может проверять содержимое ячейки под кареткой и выполнять условный переход в зависимости от результата, двигать каретку влево или вправо, стирать или устанавливать метку в текущей позиции.

Перед запуском машины нужно ввести программу, а также задать содержимое ленты и начальное положение каретки. После запуска машины возможны следующие три варианта завершения ее работы:

- работа может закончиться некорректной командой (стирание несуществующей метки или запись в уже помеченное поле);
- работа может закончиться командой «стоп»;
- работа может продолжаться бесконечно (программа «защелкнется»).

При составлении программ для машины Поста студенты используют тренажер-симулятор, разработанный преподавателем К.Ю. Поляковым [2]. Интерфейс данной программы представлен на рисунке 1.

Данный тренажер позволяет выполнять программу как в пошаговом режиме, так и непрерывно, можно регулировать скорость, при этом видно все действия, которые совершает машина.

Тренажер поддерживает две разновидности машины Поста – двоичную («классический» вариант) и троичную. В троичной машине Поста поддерживается расширенный алфавит, состоящий из пробела, символов 0 и 1. Это позволяет решать задачи, в которых числа представлены в двоичной системе счисления, как и в современных компьютерах. Поскольку студенты на практических работах используют обе версии машины Поста, то автоматизация проверки также должна поддерживать оба варианта.

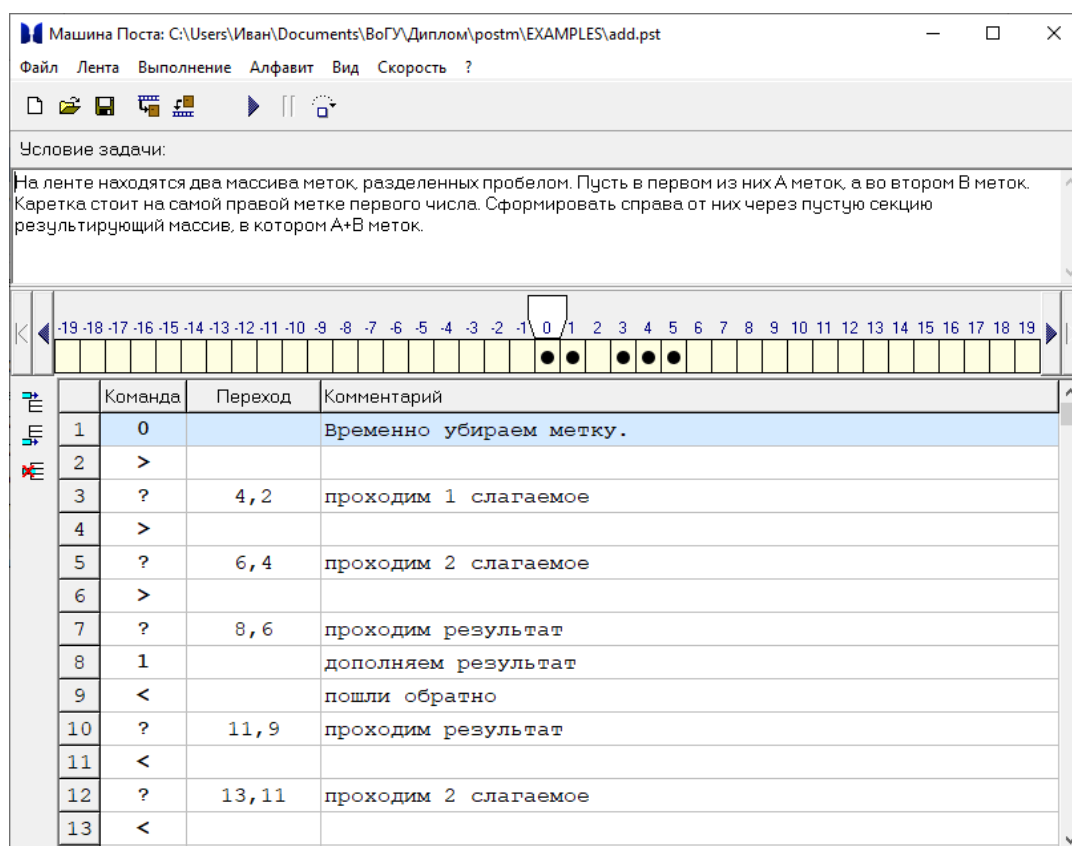


Рис. 1. Тренажер по машине Поста

Можно выделить два подхода к реализации автоматизированной проверки решений: разработка отдельного приложения либо разработка модуля для существующей проверяющей системы. Второй вариант имеет ряд преимуществ: он менее трудоемок, студенты работают с привычной системой, а их результаты преподаватель видит в привычном интерфейсе (там же, где и результаты других практических работ). В связи с этим нами был выбран именно этот вариант.

Существует целый ряд проверяющих систем, позволяющих применять собственные проверяющие модули («чекеры») – Codeforces, Яндекс.Контест, дистанционный практикум по программированию кафедры АВТ ВоГУ [3] и др. Наша разработка может применяться с любой из них, поскольку для взаимодействия проверяющего модуля с системой используются стандартные средства. Рассмотрим подробнее особенности программной реализации.

Для разработки проверяющего модуля был использован язык C++ и библиотека testlib, являющаяся стандартом де-факто для подобных целей [4]. С помощью testlib готовятся российские и международные олимпиады школьников и студентов и многие другие соревнования. Данная библиотека позволяет проверяющей программе получать от системы входные данные и студенческие решения, а также возвращать результаты проверки в систему.

Вначале нашей проверяющей программе требуется определить, какая именно машина Поста используется в задаче (двоичная или троичная). Это явно ука-

зывается преподавателем при подготовке каждой задачи.

Затем требуется указывать программе, в каком положении изначально находится каретка на ленте. Каретка может находиться в начале данных, в конце данных или в произвольной позиции внутри данных.

Далее указывается, нужно ли модулю проверять конечное положение каретки, и если надо, то указать, на каком месте должна быть каретка.

Примечание. В большинстве задач конечное положение каретки неважно, но иногда это может быть принципиально (например, если решаемая задача является частью более сложной задачи).

В модуле предусмотрено ограничение количества шагов выполнения кода. Такое ограничение имеет два применения. Во-первых, выполняемый код может зациклиться (бесконечно повторяться), и надо рано или поздно прервать его выполнение. Во-вторых, в самом задании может содержаться ограничение на количество шагов (если требуется придумать не только правильный, но еще и эффективный алгоритм).

Работа модуля начинается с получения от проверяющей системы файла с решением участника. Далее выполняется синтаксический анализ («парсинг») данного файла. В случае обнаружения ошибки в систему возвращается соответствующий числовой код и текстовое сообщение с описанием ошибки.

Примечание. Казалось бы, никаких ошибок быть не может, поскольку студенты сдают на проверку файл, сгенерированный программой-тренажером. На практике же ошибки периодически случаются, так как студенты

могут скопировать файл не полностью, скопировать не тот файл, случайно в нем что-то поменять и др.

Если ошибок не обнаружено, то от системы запрашиваются входные данные. После этого выполняется эмуляция работы машины Поста. В случае, если работа машины будет прервана из-за некорректной команды или из-за истечения количества шагов, в систему возвращается код и описание ошибки.

На последнем этапе от системы запрашивается правильный ответ, выполняется сравнение результата с ним, результат сравнения передается в систему и может быть просмотрен студентом и преподавателем.

В ходе выполнения практических работ обучающимся разрешено несколько раз сдавать на проверку свое решение – до тех пор, пока оно не будет признано верным. При этом в зависимости от настроек системы могут показываться либо не показываться входные данные, на которых решение выдает неверный ответ.

Рассмотрим примеры задач по теме «Машина Поста». Вначале приведем пример задачи для двоичной машины. На ленте записано четное число подряд идущих меток. Требуется уменьшить число меток в два раза. Результирующие метки также должны идти подряд без пропусков. Изначально каретка стоит на первой метке.

Решение этой задачи приведено на рисунке 2. Для каждой пары меток одна из них стирается, а другая

переносится в левую часть, где формируется строка с ответом.

Теперь приведем пример задачи для троичной машины Поста. На ленте записано двоичное число. Требуется получить код Грея с соответствующим порядковым номером. Код Грея представляет собой код, в котором два соседних кодовых слова различаются цифрой только в одном двоичном разряде. Последовательность кодовых слов в коде Грея начинается так: 000, 001, 011, 010, 110, 111, 101, 100...

Для решения данной задачи необходимо изучить алгоритм построения кода Грея [5] и реализовать его на машине Поста. При этом программа получается весьма компактной, она приведена на рисунке 3.

Апробация разработанного проверяющего модуля проводилась на практической работе по дисциплине «Логика и теория алгоритмов» в трех учебных группах, всего участвовал 51 студент. Для практической работы было подготовлено четыре задачи разной сложности по теме «Машина Поста». Задачи и проверяющий модуль были загружены в систему Codeforces, в которой студенты выполняют и другие работы из курса «Логика и теория алгоритмов».

Скриншот результатов проверки показан на рисунке 4. Как видим, первые три задачи успешно решили почти все. Четвертая задача была повышенной сложности. Тем не менее и с ней справились 76 % участников.

	Команда	Переход
1	<	
2	1	
3	>	
4	?	3, 5
5	0	
6	>	
7	0	
8	>	
9	?	13, 10
10	<	
11	?	10, 12
12	>	2
13	<	
14	?	13, 15
15	>	
16	1	
17	<	
18	?	19, 17
19	>	
20	0	
21	.	

Рис. 2. Пример программы для двоичной машины Поста

	Команда	Переход
1	?	0, 3, 9
2		
3	<	
4	?	0, 1, 5
5	>	
6	1	
7	<	1
8		
9	<	
10	?	0, 1, 11
11	>	
12	0	
13	<	1
14		

Рис. 3. Пример программы для троичной машины Поста

Задачи						
№	Название					
A	Деление на 2	стандартный ввод/вывод 1 с, 256 МБ			x51	
B	Пропуски	стандартный ввод/вывод 1 с, 256 МБ			x51	
C	Код Грея	стандартный ввод/вывод 1 с, 256 МБ			x50	
D	Дизъюнкция	стандартный ввод/вывод 1 с, 256 МБ			x39	
	Добавить новую задачу Добавить новые задачи из контекста					

Рис. 4. Скриншот из системы Codeforces с результатами проверки

Таким образом, опыт использования созданного модуля для автоматической проверки заданий по машине Поста подтвердил, что поставленная цель успешно достигнута: значительно снизились затраты времени, повысилось качество проверки решений.

На данный момент нами реализована автоматическая проверка решений для всех практических работ курса «Логика и теория алгоритмов». Это дает возможность создать электронный учебно-методический комплекс по данной дисциплине с теоретически неограниченным количеством участников. При этом обучающиеся могут быть уверены, что их решения будут проверены быстро и качественно.

Литература

1. Машина Поста : Википедия. Свободная энциклопедия. – URL: https://ru.wikipedia.org/wiki/Машина_Поста (дата обращения: 14.11.2023). – Текст : электронный.

2. Поляков, К. Ю. Тренажер для изучения универсального исполнителя: программа «Машина Поста» / К. Ю. Поляков. – URL: <https://kpolyakov.spb.ru/prog/post.htm> (дата обращения: 14.11.2023). – Текст : электронный.

3. Web Resource for Teaching Programming in the Form of Tournaments / I. A. Andrianov, S. U. Rzhetskiy, A. V. Rzhetskiy [et al.] // 2022 6th International Conference on Information Technologies in Engineering Education, Inforino 2022 – Proceedings : 6, Moscow, 12–15 апреля 2022 года. – Moscow, 2022. – P. 9782974. – DOI 10.1109/Inforino53888.2022.9782974.

4. Testlib: C++ library to develop competitive programming problems. – URL: <https://github.com/MikeMirzayanov/testlib> (дата обращения: 14.11.2023). – Текст : электронный.

5. Коды Грея : Викиконспекты ИТМО. – URL: https://neerc.ifmo.ru/wiki/index.php?title=Коды_Грея (дата обращения: 14.11.2023). – Текст : электронный.

I.A. Andrianov, I.A. Guzilov
Vologda State University

AUTOMATED CHECKING OF LEARNING TASKS ON TOPIC “POST MACHINE”

The article discusses the features of the implementation of the program for automatically checking students' solutions on the topic “Post Machine”. An example of a task for which automatic checking is implemented is given. The resulting software product can be used in such educational courses as “Logic and Theory of Algorithms”, “Construction and Analysis of Algorithms”.

Post machine, theory of algorithms, learning systems, e-learning.